Exact solution of the donor-limited nearest neighbor hot deck imputation problem

Jan Pablo Burgard*, Sven de Vries†, Ulf Friedrich ‡, and Dennis Kreber †§

Abstract. Data quality in population surveys suffers from missing responses. We use combinatorial optimization to create a complete and coherent data set. The methods are based on the widespread nearest neighbor hot deck imputation method that replaces the missing values with observed values from a close unit, the so-called donor. As a repeated use of donors may lead to distortions in the statistics, an additional constraint on the maximum number of donor re-uses is introduced. It is shown how this problem can be modeled as a maximum weighted b-matching problem. Based on this theoretical analysis, we implement a cost scaling, network simplex and successive shortest path algorithm for computing a globally optimal solution. These outperform the available and presently used implementations in terms of solution quality and running time. Statistical imputation is presented as a novel application of Operations Research, the available methodology is improved by a concise theoretical treatment, and fast implementations are provided.

Key words. Survey Data, Combinatorial Optimization, b-Matching, Missing Data, Nearest Neighbor Hot Deck Imputation

AMS subject classifications. 62D05, 62-07, 90C27, 90C35, 90C90

1. Introduction.

The problem of missing data is omnipresent in survey research and occurs in the two main forms of *unit non-response*, when the information for a whole unit is missing, and of *item non-response*, when single values for some units are missing [30]. Missing data cause severe problems in the statistical evaluations such as biased estimates. Moreover, incomplete data sets inhibit the direct application of estimation methods that require complete data sets, see for instance [32]. In survey statistics, it is therefore a central task to replace the missing data with appropriate values. This replacement process is called *imputation*.

In the unit non-response case, weighting adjustments are chosen to account for a possible non-response bias [13]. In the item non-response case often *single imputation* methods are used and are even more often preferred in official statistics, see [12]. In official statistics, generally a complete data set is desired for computing the necessary population figures. This is due

*Economic and Social Statistics, Trier University, 54286 Trier, Germany (burgardj@uni-trier.de)

†Mathematics, Trier University, 54286 Trier, Germany (devries@uni-trier.de)

‡Chair of Operations Research, Technical University of Munich, Arcisstr. 21, 80333 Munich, Germany (ulf.friedrich@tum.de)

§Germany, DFG-RTG Algorithmic Optimization, Trier University, 54286 Trier, Germany (kreberd@uni-trier.de)

to the fact that a complete data set automatically yields hierarchical consistent population figures, e.g., the sum of counties-figures equals the figure on the whole population.

Only one value is imputed in each missing field in single imputation. Hot deck single imputation [12] is a imputation method that maintains plausibility of records to a greater extend. The *hot deck* consists of units that are chosen to be possible donors, e.g., all units without missing values. An extensive overview of several imputation methods and how they compare to the hot deck imputation is given in [22], [2], and [26]. We focus on the *nearest neighbor hot deck imputation*, which chooses the donor for a unit with missing value(s) in terms of minimizing a predefined distance [21]. The method plays an important role in official statistics and was used in censuses conducted in Canada, the United Kingdom, Italy, Brazil, the USA, New Zealand, Australia and Switzerland [3] and recently in Germany [11].

In practice it may be desirable to limit the amount of times a donor may be re-used in a hot deck. In [19] the effect of limiting the number of times a donor may be re-used is studied and concluded that in general a reduced variance can be expected when taking each donor the least possible time. The authors of [18] combine the nearest neighbor aspect with a donor limitation because this imputation method is especially prone to an over-usage of donors. They examine the effects and find that a donor-limited nearest neighbor hot deck imputation is especially advantageous on data sets with a high percentage of missing values. They advise to minimize the sum of *all* distances between donors and recipients. Following up on this, [16], [17] state that this minimization problem can be formulated as a transportation problem. The author proposes two heuristic algorithms that, however, do not guarantee the quality of the computed solution.

In contrast, we find a globally optimal solution to the donor-limited problem with an arbitrary donor-specific integer-limit. Based on a graph theoretic model for the problem, we develop exact combinatorial algorithms and compare their speed and stability with respect to the total number of units in the data set.

Contribution and Structure. The nearest neighbor hot deck imputation is of great significance in official statistics and an additional donor-limit introduces a beneficial extension to the imputation methodology. Yet, to our knowledge, work on this topic has only been conducted by [16], [17] and [18]. We propose mathematical precise methods for finding a donor-limited nearest neighbor hot deck imputation. Our contributions include the development of models on which we can apply combinatorial algorithms to find globally optimal solutions to the donor-limited hot deck imputation problem. In contrast to the work of [16], [17], we propose using *exact* methods instead of applying heuristics.

The presented model gives a more flexible control over possible imputations. Donor limits can be chosen independently for each donor unit and specific imputation pairs can be ex- or included from the model, which allows more control over desired statistical properties.

We show that even though we calculate the global optimum, computation times stay at a minimum. In fact, our implementations are considerably faster than the `R`-Package *HotDeckImputation* [18]. Furthermore, we find that the successive shortest path algorithm often benefits from the structure of the problem graph.

In Section 2, a concise mathematical formulation of the nearest neighbor hot deck imputation is presented. It includes the definitions of missing patterns, distances and relations, and the formulation of the mathematical problem. Then, the optimal solution to the problem is derived in Section 3. In the last section, numerical results based on the publicly available data set AMELIA [24] are presented. AMELIA is a synthetic and realistic household data set. The paper concludes with a summary and outlook.

2. Nearest Neighbor Hot Deck Imputation.

Before setting up the optimization problem, we define the donors, receivers and the distances between them in a formal way. Let $X_1, \dots, X_Q$ be subsets of $\mathbb{R}$, the so-called *attributes*. For $N \in \mathbb{N}$ a *population* $\Omega \in \prod_{j=1}^{Q} X_j^N$ of *size* N is defined as an $N \times Q$ matrix whose values in each column j are elements of the attribute X_j. Even though we allow categorical and dichotomous variables (see A), we assume without loss of generality that they are encoded as values in $\mathbb{R}$.

Definition 2.1. *Let $n \leq N$ and $q \leq Q$. A submatrix $S \in \mathbb{R}^{n \times q}$ of Ω is called a complete sample. A sample is a matrix $\mathscr{S} \in (\mathbb{R} \cup \{\texttt{NA}\})^{n \times q}$ in which some elements of S are replaced with the value `NA` for "not available".*

Each row $(u_{i1}, \dots, u_{iq})$ of a sample $\mathscr{S}$ is called a *unit*. We also write $(u_{i1}, \dots, u_{iq}) \in \mathscr{S}$ even though $\mathscr{S}$ is not a set in the technical sense. We also omit the row index and just write $(u_1, \dots, u_q)$ or u to denote a unit. A unit u is called a *respondent* if $u_j \neq \texttt{NA}$ for every j. Otherwise, u is called a *non-respondent*.

Definition 2.2. *Let $\mathscr{S}$ be a sample. A relation $\prec$ on $\mathscr{S} \times \mathscr{S}$ is called a receiver-donor relation if $(u_1, \dots, u_q) \prec (v_1, \dots, v_q)$ implies that u is a non-respondent and that $v_i \neq \texttt{NA}$ for all i with $u_i = \texttt{NA}$. The set $A_\prec(\mathscr{S}) := \{(v, u) \in \mathscr{S}^2 : u \prec v\}$ describes all pairs of receivers u and donors v. The donor set $D_\prec$ and receiver set $R_\prec$ are defined as follows:*

$$R_\prec := \{u \in \mathscr{S} : u \prec v \text{ for some } v \in \mathscr{S}\}, \tag{2.1}$$

$$D_\prec := \{v \in \mathscr{S} : u \prec v \text{ for some } u \in \mathscr{S}\}. \tag{2.2}$$

In unambiguous cases the index $\prec$ will be omitted in the above definitions and we write $A(\mathscr{S})$, R, and D instead. The reasoning behind the concept of a receiver-donor relation is of course that $v \in \mathscr{S}$ is a potential donor for $u \in \mathscr{S}$ whenever $u \prec v$ holds. Generally, there is more than one possible donor for a given receiver and one has to choose the in a certain sense *best* donor while respecting possible constraints. Finding an appropriate receiver-donor relation is important. It can for instance be advantageous to choose it in such a way that all receiver-donor pairs contain enough mutual information, which facilitates a meaningful comparison. On the other hand, if a receiver-donor relation is chosen too restrictive, the set of possible donors might be too small and the imputation problem is rendered infeasible. A compromise solution in this case considers receiver-donor pairs where the missing values of both the donor and receiver are at least mutually complementary.

If a data set has few missing values and more respondents than non-respondents, one can often define a receiver-donor relation by considering only respondents as donors and non-respondents as receivers. This relation ensures that the quality of each imputation can be evaluated consistently. We call this receiver-donor relation the *respondent-non-respondent relation*. Formally, the relation is defined by setting $u \prec v$ if and only if u is a non-respondent and v is a respondent.

Possible receiver-donor pairs become comparable by introducing a distance.

Definition 2.3. *Let $\mathscr{S}$ be a sample and $\prec$ a receiver-donor relation. We call a symmetric mapping*

$$d : A_{\prec}(\mathscr{S}) \cup \{(u, v) : (v, u) \in A_{\prec}(\mathscr{S})\} \to \mathbb{R}_+$$

a distance of $\mathscr{S}$.

Common distances in the missing data context are the Gower distance [10], the Mahalanobis distance [23], and distances based on the *predictive mean* as discussed by [2]. As we use the Gower distance for the computations in Section 5, we review its definition in A.

3. Formulation of the Optimization Problem. We have provided the necessary framework for the formulation of the imputation problem for a given sample $\mathscr{S}$ and a receiver-donor relation $\prec$ as a mathematical optimization problem. The objective is the minimization of the sum over all distances of all imputation pairs, while respecting an upper bound on each donor's maximal contribution. The upper bound for each donor v is denoted by $b_v \in \mathbb{Z}_+$ and the vector of all upper bounds by b. Usually, all donations are bounded by the same value, but our model allows to choose individual donation constraints.

We model the problem as a graph problem by considering the union of the sets R and D, defined in (2.1) and (2.2), as the vertex set V of a graph $G = (V, E)$. Although in general R and D are not disjunct, we simplify the construction without loss of generality in such a way that each element in R and D is treated as a unique entity and thus as a single node in the graph. Formally, this is achieved by extending the units with an $(q+1)$-th binary variable indicating if the unit is in the set R or D and taking the union of the extended units. That means a receiver unit u is replaced by an auxiliary unit $(u, 1)$ and a donor unit v is replaced by $(v, 0)$. If u is a receiver and a donor, then we use two auxiliary units $(u, 1)$ and $(u, 0)$ to construct the sets R and D.

We write $V = R \cup D$ for the disjunct union and obtain $|V| = |R| + |D|$, where $|\cdot|$ is the cardinality of a set. The set of edges E of the graph is defined by $E \coloneqq \{\{u, v\} \in \mathscr{S}^2 : u \prec v\}$. Following the common notation, we let $m \coloneqq |E|$ and $n \coloneqq |V|$.

The graph defined above is by construction bipartite, i.e., there are only edges connecting an element of R with an element of D but not between two elements of the same set. We complete the construction by assigning distances d to each edge and an upper bound $b_v \in \mathbb{Z}_+$ to each donor $v \in D$.

Now, the imputation problem corresponds to the problem of finding a subset M of E in which each $u \in R$ is covered by an edge of M, each $v \in D$ is an end-node of maximally $b_v \in \mathbb{Z}_+$ edges and M is minimal with respect to the distance d. Choosing such an M is achieved by introducing indicator variables x, i.e., $x_e = 1$ if and only if the edge e is in M. This yields the linear integer optimization problem (HDI1).

$$
\begin{aligned}
\min \quad & \sum_{e \in E} d(e)\, x_e \\
\text{s.t.} \quad & \sum_{e \in \delta(u)} x_e = 1 && \forall u \in R \\
& \sum_{e \in \delta(v)} x_e \leq b_v && \forall v \in D \\
& x_e \in \{0, 1\} && \forall e \in E,
\end{aligned}
\tag{HDI1}
$$

where $\delta(u)$ denotes the set of edges covering u. Throughout our presentation, we assume that the instances of Problem (HDI1) and its equivalent reformulations have at least one feasible solution.

We first consider the case $b = \mathbf{1}$. The resulting problem loosely resembles a perfect minimum weight matching. However, since in practice we have more possible donors than receivers,

a perfect matching does not exist. Therefore, rather than requiring a perfect matching, we compute a minimum-weight matching M, such that M is a maximum-cardinality matching.

In [29, Chapter 17] a modified Hungarian method for solving the perfect minimum-weight matching problem is proposed. Examining the method closely shows that the algorithm stops when a maximum-cardinality matching is found. Furthermore, such a matching is extreme [29, Theorem 17.2], i.e., it has the minimum weight under all matchings with the same size. Thus, the Hungarian method can be used to solve (HDI1) when fixing $b = \mathbf{1}$. As a matter of fact, Schrijver's implementation of the Hungarian method can be regarded as a successive shortest path algorithm on a minimum-cost flow problem.

Although this observation only holds for the standard matching case $b = \mathbf{1}$, it is our goal to extend the described idea to arbitrary b. As seen in Section 4, an analogous approach is indeed applicable for the general case and the successive shortest path algorithm turns out to be a suitable choice for solving (HDI1), see Section 5.

We formulate the problem as a maximum-weighted b-matching problem, as defined by [29, Chapter 31]. This allows a broader perspective for theoretical examination, available solution methods, and problem complexities.

The transformation is straightforward and follows the idea in [25, pp. 610], see also B.1. We obtain the maximum-weighted b-matching problem (HDI2) as an equivalent formulation for (HDI1).

$$
\begin{aligned}
\max \quad & \sum_{e \in E} w(e)\, x_e & & \\
\text{s.t.} \quad & \sum_{e \in \delta(u)} x_e \leq 1 & & \forall u \in R \\
& \sum_{e \in \delta(v)} x_e \leq b_v & & \forall v \in D \\
& x_e \in \mathbb{Z}_+ & & \forall e \in E,
\end{aligned}
\tag{HDI2}
$$

with $w(e) \coloneqq |R| \max_{f \in E} d(f) + 1 - d(e)$.

4. Solving the Nearest Neighbor Hot Deck Imputation Problem. A straightforward approach for the solution of (HDI2), and equivalently (HDI1), is to find a solution for the LP relaxation of the problem. Indeed, it is well-known that the node-edge incidence matrix of a bipartite graph, like in (HDI2), is totally unimodular. Therefore, every optimal solution of the linear relaxation of (HDI2) is an integer point. The proof for the above classical result is for instance given in Chapter 3 of [4].

When dealing with small values for b_v, one could also introduce b_v copies of each donor node $v \in D$ and solve a weighted matching problem on the resulting graph. However, this is not a

polynomial time transformation and solving such a matching problem becomes computational expensive, even for small donor limits.

We are interested in developing more adapted solvers for the imputation problem. As it is equivalent to a maximum-weighted b-matching problem, there exist efficient algorithms to solve it. The algorithm presented in [8] takes $\mathcal{O}(m^2 \log n \log\|b\|_\infty)$ steps to solve a weighted b-matching problem. However, as the expression m^2 dominates this bound in dense graphs that typically occur in nearest neighbor hot deck imputation problems, it is preferable to reduce the worst-case number of iterations over all edges. This is possible by transforming (HDI1) to a capacitated minimum cost flow problem. In practical applications, the cost scaling algorithm, developed independently by [28] and [5], and the network simplex algorithm are the most efficient algorithms to solve minimum cost flow problems [20]. The cost scaling algorithm was further enhanced by [9] to a runtime of $\mathcal{O}(nm \log(n^2/m) \log(nC))$ and the network simplex algorithm can find an optimal solution in $\mathcal{O}(nm \log(n) \log(nC)$ with $C = \|d\|_\infty$, see [31]. The disadvantage of both approaches is that they require integer distances, which is generally not the case for imputation problems. Thus, before applying any of the two algorithms, it is required to multiply the distances by a large factor and round them to the next integer. This has negative effects on the runtime because the logarithm of the maximal distance is a factor in the runtime bound.

We propose using the successive shortest path algorithm on the imputation problem published independently by [15], [14] and [6]. Although it generally runs only in pseudo-polynomial time, we show that it performs better on the imputation problem and reaches a worst-case runtime of $\mathcal{O}(|R|(m + n \log n))$. Moreover, the algorithm permits non-integral distances. As noted before, the successive shortest path approach is also in line with methods for bipartite perfect minimum-cost matchings, which further motivates its use.

4.1. Transformation to a capacitated minimum cost flow problem. Following a standard approach for solving bipartite b-matching problems, we create a new source node s that contains all donations and transform the imputation problem into a capacitated minimum cost flow problem. This transformation is possible because the imputation graph is bipartite and fails for general graphs.

The directed graph (digraph) with vertex set $V_1 := R \cup D \cup \{s\}$ and arc set $A_1 := A_\prec(\mathscr{S}) \cup \{(s, v) : v \in D\}$ is constructed. Furthermore, we define for $v \in D$ and $u \in R$ the weights

$$d_1((v,u)) := \begin{cases} d(\{v,u\}), & \text{if } (v,u) \in A_\prec(\mathscr{S}), \\ 0, & \text{otherwise.} \end{cases}$$

On this digraph we can set up the minimum cost flow problem.

$$
\begin{aligned}
\min \quad & \sum_{a \in A_1} d_1(a) x_a & & \\
\text{s.t.} \quad & \sum_{a \in \delta^{\text{out}}(v)} x_a - \sum_{a \in \delta^{\text{in}}(v)} x_a = 0 & & \forall v \in D \\
& - \sum_{a \in \delta^{\text{in}}(u)} x_a = -1 & & \forall u \in R \\
& \sum_{a \in \delta^{\text{out}}(s)} x_a = |R| & & \\
& x_{(v,u)} \leq 1 & & \forall v \in D, u \in R \\
& x_{(s,v)} \leq b_v & & \forall v \in D,
\end{aligned}
\tag{HDI3}
$$

where $\delta^{\text{in}}(v)$ denotes the set of arcs ending in v and and $\delta^{\text{out}}(v)$ denotes the set of arcs starting in v.

In each iteration the successive shortest path algorithm reduces the node imbalance of two vertices by at least 1 for each of the two vertices, see [1, Chapter 9] and definitions therein. Since the total, absolute imbalance of (HDI3) is $2|R|$, it requires $|R|$ iterations. In each iteration a shortest path is searched in time $\mathcal{O}(m + n \log n)$ using Dijkstra's algorithm with Fibonacci heaps [29]. Therefore, the successive shortest path algorithm takes $\mathcal{O}(|R|(m + n \log n))$ time to find an optimal solution of (HDI3).

5. Computational Study.

We have modeled the generalized hot deck imputation problem with donor constraints in a mathematically concise way and presented how it can be solved as a combinatorial optimization problem. In this section, we show that the theory also translates to powerful algorithms for practical problem solving.

Algorithms. Our implementations comprise the network simplex (NSX), cost-scaling (CS), and successive shortest path (SSP) algorithm. For all implementations the open source `C++` LEMON graph library [7] is used. While NSX and CS algorithms are provided by LEMON, we implemented the SSP algorithm using LEMON's data structures and its implementation of Dijkstra's algorithm. Since both the network simplex and the cost-scaling approach require integer distances on the arcs, we multiply all distances with a sufficiently large factor, in our case 10^8, and round with the *round half up* rule.

We assess the quality of the above algorithms in comparison to the heuristic methods matrix minimization (MMIN) and Vogel approximation method (VAM) available in the `R` package *HotDeckImputation* [18]. The MMIN method is a greedy algorithm and Vogel's approximation is a heuristic for transportation problems proposed by [27]. Both methods

are mentioned as adequate tools by [16]. The package *HotDeckImputation* also contains a network simplex implementation which, however, does not consistently find optimal solutions and every so often failed to terminate. To our knowledge, *HotDeckImputation* is the only available solver for the donor-limited hot deck imputation problem so far.

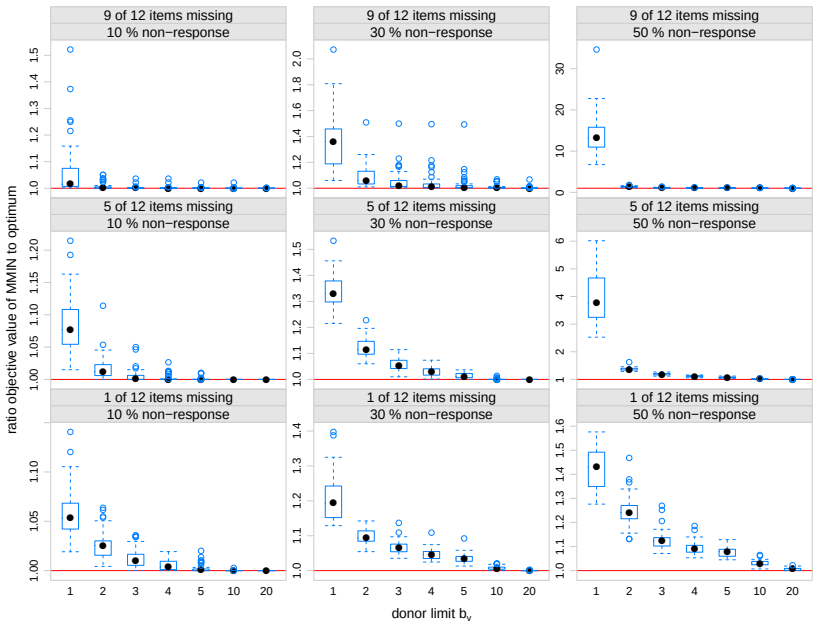

Figure 5.1. *Ratios of the objective value of the matrix minimum method (MMIN) to the optimal value. Each boxplot consist of 49 independent samples of size 2000.*

Simulation setup. The imputations are calculated on the synthetic data set AMELIA [24] using 12 variables of categorical, ordered, and continuous scale. We consider sample sizes of $\texttt{size} = \{1000, 2000, 3000, 5000\}$ units. For each sample size, 49 simple random samples without replacement are drawn. Within of each of these samples, $\texttt{unit} = \{10\%, 30\%, 50\%\}$ of the units are randomly selected to contain missings. The number of missing items is then chosen to be $\texttt{item} = \{1, 5, 9\}$ out of the total 12 items. The items to be missing are again drawn randomly. This can be regarded as an MCAR pattern, as discussed by [22]. In total, we create $|\texttt{unit} \times \texttt{item}| = 15$ scenarios for each of the $|\texttt{size}| * 49 = 196$ samples. For each scenario in each sample the optimal solution for the donor limited hot deck nearest imputation problem is computed using the donor-limits $b_v = 1, \ldots, 5, 10, 20$. The optimization problems are generated using the respondent-non-respondent relation and the Gower distance.

Quality of solutions. The imputation quality relies on the objective value and minimizing the sum over all distances implies improved statistical qualities of the imputed data set, see [16].

We compare the final objective values of MMIN with the optimal solution to the imputation problem. In many samples the objective value of MMIN is much higher than the optimal value (Figure 5.1), up to a factor of 35. The approximation quality of MMIN is especially poor for the restrictive situation with a low donor-limit of $b_v = 1$. The greedy approximation becomes better for higher donor limits. This is plausible since donor limits pose no restriction if chosen sufficiently large. Thus, a greedy method finds an optimal solution in this case. The same patterns can be observed in Table 5.1 when comparing median objective values. Approximation quality improves when increasing b_v, but is very poor for $b_v = 1$. For limits above 5 the approximation quality of MMIN becomes tolerable.

b_v	1	2	3	4	5	10	20
CS	4.70	2.99	2.75	2.66	2.63	2.62	2.62
SSP	4.70	2.99	2.75	2.66	2.63	2.62	2.62
NSX	4.70	2.99	2.75	2.66	2.63	2.62	2.62
MMIN	19.88	4.05	3.29	2.94	2.79	2.64	2.63
VAM	16.01	-	-	-	-	-	-

Table 5.1

Median objective value for the scenario with 50% missing units, 5 items missing out 12, and sample size 2000.

Likewise, we compare the objective values produced by VAM to the optimal values in Table 5.1. The median of the objective values returned by VAM for the case $b_v = 1$ is three times higher than the median of the optimal values. We can not compare VAM to larger donor limits since the implementation provided in *HotDeckImputation* fails on all instances with $b_v > 1$ and non-respondents rates of less then 50%, i.e., if a perfect b-matching does not exist. We suspect that the code has to be adapted to handle excessive supply in the transformation of the transportation problem. In comparison to the MMIN method, VAM has a better approximation quality, which was also observed by [16].

In conclusion, both methods provided by *HotDeckImputation* do generally not find the optimum and return objective values which can be significantly worse than optimal value. The key assumption of the nearest neighbor hot deck imputation is that smaller distances imply an improved imputation quality. Thus, lower objective values lead to enhanced estimations of missing values and optimal solvers should be used whenever available. Furthermore, VAM is a rather uncommon method and to our understanding does not yield any advantages over

more sophisticated algorithms. Moreover, approximation qualities for this method are not guaranteed, which makes its usage even more questionable.

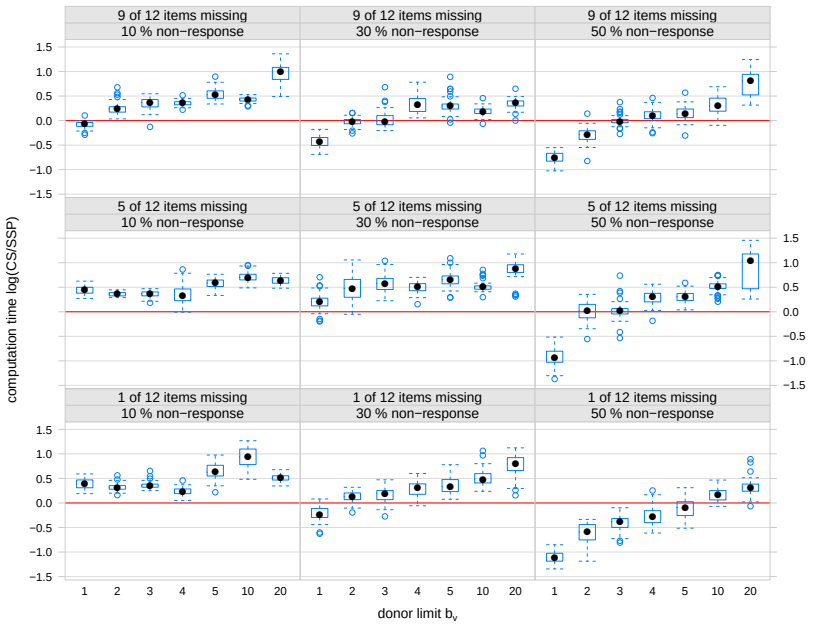

Figure 5.2. *Log Ratios of measured computation times between the implementation of a successive shortest path algorithm (SSP) and the LEMON cost-scaling algorithm (CS). Each boxplot consist of 49 independent samples of size 2000.*

Computation times. In Figure 5.2 the log ratio of the computation times of the CS to the SSP algorithm is depicted for the sample size 2000. The computation times of the CS algorithm are higher (non-negative log ratio) in most cases However, if the imputation problem is restrictive, i.e., a high percentage of units are non-respondents and the donor limit is low, CS performs faster. This behavior is more pronounced for larger sample sizes (see Figure 5.3), where SSP is never faster in the scenario with 50% missings and 1 of 12 items missing. Moreover, it can be observed that the performance of NSX is very similar to the performance of CS. Nevertheless, SSP is the superior choice in most settings, whereas CS or NSX should be considered when the imputation problem has few donors or is overly restrictive. For the scenario of sample size 2000, 50% missing units, and 5 out of 12 items missing we present the median computation times for the five algorithms SSP, CS, NSX, VAM, and MMIN in Table 5.2.

Use of inexact heuristics might be justified when exact algorithms turn out to be too resource-consuming. However, we could not observe a considerable performance advantage for the *HotDeckImputation* implementations in Figure 5.4 and Table 5.2. In conclusion, the

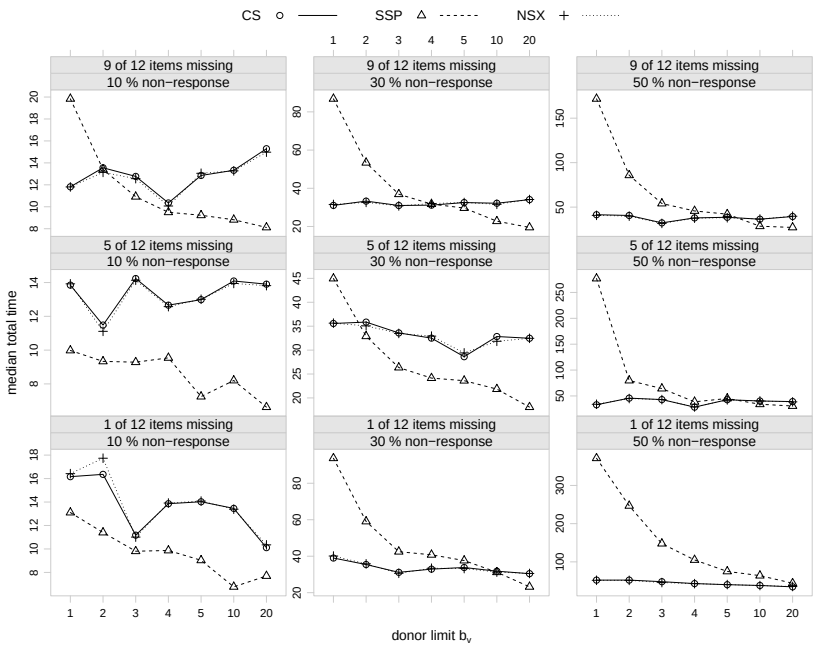

Figure 5.3. *Median computation time in seconds of the LEMON cost-scaling algorithm (CS), the successive shortest path algorithm (SSP), and the network simplex (NSX). Each point is the median computation time over the 49 independent samples of size 5000.*

heuristic methods do neither provide a sufficient approximation quality nor do they perform better in terms of running times.

b_v	1	2	3	4	5	10	20
CS	7.82	6.89	4.74	6.22	6.25	6.02	8.05
SSP	20.49	6.68	4.76	4.76	4.64	3.57	3.16
NSX	7.85	6.90	4.79	6.34	6.19	5.98	8.02
MMIN	8.96	9.80	10.73	10.47	11.05	8.97	13.45
VAM	8.01	-	-	-	-	-	-

Table 5.2

Median computation time for the scenario with 50% missing units, 5 items missing out of 12, and sample size 2000.

6. Summary. Nearest neighbor hot deck imputation is an important and commonly used tool in Data Science. [19] and [18] show that limiting the number of times a donor may be re-used can improve the imputation quality. However, the methodology for solving a donor-limited nearest neighbor hot deck imputation problem has not been examined thoroughly until now. We have closed this gap by modeling the imputation problem as a maximum-weighted

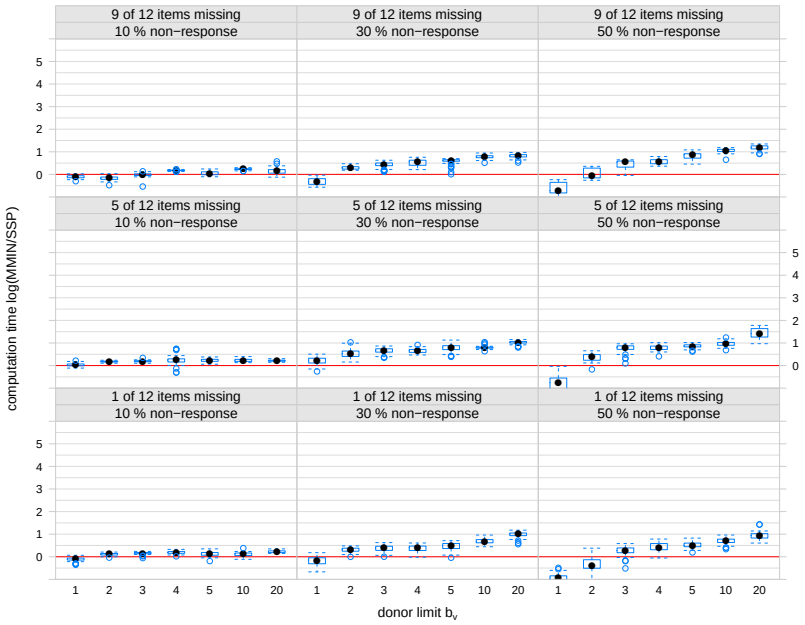

Figure 5.4. *Log Ratios of measured computation times between the implementation of a successive shortest path algorithm (SSP) and the matrix minimum method (MMIN) from the* HotDeckImputation *R-Package. Each boxplot consist of 49 independent samples of size 2000.*

b-matching, formulating a graph problem, and computing donor-limited nearest neighbor hot deck imputations with the help of exact combinatorial algorithms.

We have tested several algorithms on the open data set AMELIA for multiple scenarios. The computational study shows that our algorithms compute solutions that are superior to those of the priorly available solvers. Comparing the numerical performance of the algorithms, our implementation of the successive shortest path algorithm outperforms the other algorithms in most interesting cases. If the percentage of missing units is particularly high and donor limits are low, the usage of CS or NSX is more advantageous.

The presented graph theoretic approach gives both mathematically grounded, theoretical insights as well as fast algorithms for the computation of a global solution to the donor-limited nearest neighbor hot deck imputation problem in practice.

Acknowledgements. This research was supported within the research training group 2126 *Algorithmic Optimization* (ALOP) founded by the German Research Foundation DFG.

REFERENCES

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms and Applications*, Prentice-Hall, Upper Saddle River, 1993.

[2] R. R. Andridge and R. J. A. Little, *A review of hot deck imputation for survey non-response*, International Statistical Review, 78 (2010), pp. 40–64.

[3] M. Barnier, *Edit and imputation for the 2006 Canadian census*, in Conference of European statisticians, 2005.

[4] D. Bertsimas and R. Weismantel, *Optimization over Integers*, Dynamic Ideas, Belmont, 2005.

[5] R. G. Bland and D. L. Jensen, *On the computational behavior of a polynomial-time network flow algorithm*, Mathematical Programming, 54 (1992), pp. 1–39.

[6] R. G. Busacker and P. J. Gowen, *A procedure for determining a famliy of minimum-cost network flow patterns*, tech. report, Operations Research Office, Johns Hopkins University, 1960.

[7] B. Dezső, A. Jüttner, and P. Kovács, *LEMON – An open source C++ graph template library*, Electronic Notes in Theoretical Computer Science, 264 (2011), pp. 23–45. See also http://lemon.cs.elte.hu/trac/lemon.

[8] H. N. Gabow, *An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems*, in Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing, 1983, pp. 448–456.

[9] A. Goldberg and R. Tarjan, *Solving minimum-cost flow problems by successive approximation*, in Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing, 1987, pp. 7–18.

[10] J. C. Gower, *A general coefficient of similarity and some of its properties*, Biometrics, 27 (1971), pp. 857–871.

[11] S. Grunwald and A. Krause, *Umgang mit fehlenden Angaben in der Gebäude- und Wohnungszählung 2011*, Statistisches Bundesamt, Wirtschaft und Statistik, (2014).

[12] D. Haziza and G. Kuromi, *Handling item non response in surveys*, Case Studies in Business, Industry And Government Statistics, 1 (2014), pp. 102–118.

[13] D. Haziza and É. Lesage, *A discussion of weighting procedures for unit nonresponse*, Journal of Official Statistics, 32 (2016), pp. 129–145.

[14] M. Iri, *A new method of solving transportation-network problems*, Journal of the Operations Research Society of Japan, 3 (1960), pp. 27–87.

[15] W. S. Jewell, *New methods in mathematical programming – optimal flow through networks with gains*, Operations Research, 10 (1962), pp. 476–499.

[16] D. W. Joenssen, *Donor limited hot deck imputation: A constrained optimization problem*, in Studies in Classification, Data Analysis, and Knowledge Organization, vol. 48, Springer, Berlin, Heidelberg, 2015, pp. 319–328.

[17] D. W. Joenssen, *Hot-Deck-Verfahren zur Imputation fehlender Daten: Auswirkungen des Donor-Limits*, PhD thesis, Technische Universität Ilmenau, 2015.

[18] D. W. Joenssen and U. Bankhofer, *Donor limited hot deck imputation: effects on parameter estimation*, Journal of Theoretical and Applied Computer Science, 6 (2012), pp. 58–70.

[19] G. Kalton and L. Kish, *Two efficient random imputation procedures*, in Proceedings of the Survey Research Methods Section, 1981, pp. 146–151.

[20] P. Kovács, *Minimum-cost flow algorithms: an experimental evaluation*, Optimization Methods and Software, 30 (2015), pp. 94–127.

[21] A. Kowarik and M. Templ, *Imputation with the R package VIM*, Journal of Statistical Software, 74 (2016), pp. 1–16.

[22] R. J. A. Little and D. B. Rubin, *Statistical Analysis with Missing Data*, John Wiley & Sons, Hoboken, 2nd ed., 2002.
[23] P. C. Mahalanobis, *On the generalized distance in statistics*, Proceedings of the National Institute of Sciences (Calcutta), 2 (1936), pp. 49–55.
[24] H. Merkle, J. P. Burgard, and R. Münnich, *The AMELIA Dataset – A Synthetic Universe for Reproducible Research*, in Deliverable 23.1: Case Studies, Y. G. Berger, J. P. Burgard, A. Byrne, A. Cernat, C. Giusti, P. Koksel, S. Lenau, S. Marchetti, H. Merkle, R. Münnich, I. Permanyer, M. Pratesi, N. Salvati, N. Shlomo, D. Smith, and N. Tzavidis, eds., vol. WP23 – D23.1, 2016.
[25] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, John Wiley & Sons, New York, 1988.
[26] A. Olinsky, S. Chen, and L. Harlow, *The comparative efficiency of imputation methods for missing data in structural equation modeling*, European Journal of Operational Research, 151 (2003), pp. 53–79.
[27] N. V. Reinfeld and W. R. Vogel, *Mathematical Programming*, Prentice-Hall, 1958.
[28] H. Röck, *Scaling techniques for minimal cost network flows*, in Discrete Structures and Algorithms – Proceedings of the 5th Conference on Graphtheoretic Concepts in Computer Science, 1980, pp. 181–191.
[29] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*, Springer, Berlin, 2003.
[30] J. Stack, *The analysis of survey data*, European Journal of Operational Research, 81 (1995), pp. 1–16.
[31] R. E. Tarjan, *Dynamic trees as search trees via euler tours, applied to the network simplex algorithm*, Mathematical Programming, 78 (1997), pp. 169–177.
[32] J. Zhang and H. Aytug, *Comparison of imputation methods for discriminant analysis with strategically hidden data*, European Journal of Operational Research, 255 (2016), pp. 522–530.

Appendix A. Gower Distance.

Following [10], a popular method to measure proximity is the so-called *Gower distance*. It differentiates between nominal and ordinal continuous attributes and measures normalized, absolute differences between single variables. The total sum of these differences is then called the Gower distance. This idea is rendered precisely in the next two definitions. While the first Definition introduces pointwise distances for dichotomous (i), quantitative (ii), and qualitative (iii) variables, the subsequent definition combines the pointwise values in the actual Gower distance. A *quantitative* variable can be ordered while a *qualitative* variable is categorical. A *dichotomous* variable has two states, the + state and the − state. From + unambiguous information can be deduced, whereas − generally resembles several possible instances. For example the question "Are you married?" does not specify the possibilities single, divorced or widowed when answered with "no".

Definition A.1. *Let $\mathscr{S}$ be a sample with q columns and $u, v \in \mathscr{S}$. The pointwise distance $g_j(u, v)$ between the variables of two units u, v is given by:*

i) If the j-th variable is dichotomous, then

$$g_j(u,v) \coloneqq \begin{cases} 0, & \text{if } u_j = v_j = +, \\ 1, & \text{otherwise.} \end{cases}$$

ii) If the j-th variable is quantitative, then

$$g_j(u,v) \coloneqq \begin{cases} 0, & \text{if } u_j = \texttt{NA} \text{ or } v_j = \texttt{NA}, \\ \frac{|u_j - v_j|}{(M_j - m_j)}, & \text{otherwise.} \end{cases}$$

where $M_j \coloneqq \max\{w_j : w \in \mathscr{S}\}$ and $m_j \coloneqq \min\{w_j : w \in \mathscr{S}\}$.

iii) If the j-th variable is qualitative, then

$$g_j(u,v) \coloneqq \begin{cases} 0, & \text{if } u_j = v_j, \\ 1, & \text{otherwise.} \end{cases}$$

We say that two values u_j and v_j are *comparable* if none of them is missing, i.e., $u_j \neq \texttt{NA}$ and $v_j \neq \texttt{NA}$, or in the case of dichotomous values if at least one of the two is $+$. The pointwise definitions above cover the cases when values are not comparable, but the associated distances are not considered in the final summation.

Definition A.2. *Let μ be defined by*

$$\mu_j(u,v) \coloneqq \begin{cases} 1, & \text{if } u_j \text{ is comparable to } v_j, \\ 0, & \text{else.} \end{cases}$$

Then, the Gower distance *g is defined by*

$$g(u,v) \coloneqq \frac{\sum_{j=1}^{q} \mu_j(u,v) g_j(u,v)}{\sum_{j=1}^{q} \mu_j(u,v)}$$

for all u, v with $\mu_j(u,v) = 1$ for at least one j.

In short, the Gower distance is defined as the mean over all $g_j(u,v)$ for which the units u, v are comparable. The results in Sections 4 and 5 are formulated in terms of the Gower distance as it is widely used in practice. However, our results remain valid for all distance choices.

Appendix B. Transformation to b-matching. Every perfect minimum weighted matching problem can be formulated as a maximum weight matching problem [25, pp. 610]. This idea can be extended straightforwardly.

Proposition B.1. *Let $G = (U_1 \cup U_2, E)$ be a bipartite graph with edge weights $w : E \to \mathbb{R}$, $A \in \mathbb{R}^{m \times n}$ If the problem*

$$\begin{aligned} \min \quad & \sum_{e \in E} -w(e)x_e \\ \text{s.t.} \quad & \sum_{e \in \delta(u)} x_e = 1 \qquad \forall u \in U_1 \\ & Ax \leq b \\ & x_e \in \{0,1\} \quad \forall e \in E \end{aligned} \tag{B.1}$$

is feasible, then every optimal solution of

$$\begin{aligned} \max \quad & \sum_{e \in E} (w(e) + \theta)x_e \\ \text{s.t.} \quad & \sum_{e \in \delta(u)} x_e \leq 1 \qquad \forall u \in U_1 \\ & Ax \leq b \\ & x_e \in \{0,1\} \quad \forall e \in E \end{aligned} \tag{B.2}$$

is also an optimal solution of (B.1).