Journal of the American Statistical Association

ISSN: 0162-1459 (Print) 1537-274X (Online) Journal homepage: https://www.tandfonline.com/loi/uasa20

Optimal Matching for Observational Studies

Paul R. Rosenbaum

To cite this article: Paul R. Rosenbaum (1989) Optimal Matching for Observational Studies, Journal of the American Statistical Association, 84:408, 1024-1032

To link to this article: https://doi.org/10.1080/01621459.1989.10478868

Optimal Matching for Observational Studies

PAUL R. ROSENBAUM*

Matching is a common method of adjustment in observational studies. Currently, matched samples are constructed using greedy heuristics (or "stepwise" procedures) that produce, in general, suboptimal matchings. With respect to a particular criterion, a matched sample is suboptimal if it could be improved by changing the controls assigned to specific treated units, that is, if it could be improved with the data at hand. Here, optimal matched samples are obtained using network flow theory. In addition to providing optimal matched-pair samples, this approach yields optimal constructions for several statistical matching problems that have not been studied previously, including the construction of matched samples with multiple controls, with a variable number of controls, and the construction of balanced matched samples that combine features of pair matching and frequency matching. Computational efficiency is discussed. Extensive use is made of ideas from two essentially disjoint literatures, namely statistical matching in observational studies and graph algorithms for matching. The article contains brief reviews of both topics.

KEY WORDS: Graph algorithms; Network flow; Propensity score; Statistical computing.

1. INTRODUCTION

1.1 Two Literatures on Matching

There are two essentially disjoint literatures on matching. The first is the statistical literature on the construction of matched samples for observational studies. The second is the literature in discrete mathematics, computer science, and operations research on matching in graphs and networks. This article uses ideas from the second literature as they relate to problems in the first.

The article is organized as follows. Section 1.2 reviews certain statistical aspects of matching in observational studies. Section 1.3 discusses a tangible example that illustrates the difference between an optimal matching and a matching constructed by the greedy heuristics that are currently used by statisticians. The key point is that two or more treated units may have the same control as their best match, and conventional heuristics resolve this bottleneck in an arbitrary way, typically yielding a suboptimal match, that is, a matched sample that could be improved with the data at hand. Greedy and optimal matching are compared in Section 1.4. Relevant network flow theory is briefly reviewed in Section 2, with extensive references. Network flow methods are used to solve a series of statistical matching problems in Section 3, including matching with multiple controls, matching with a variable number of controls, and balanced matching. Computational considerations are discussed in Section 4.

1.2 Constructing Matched Samples in Observational Studies: A Short Review

An observational study is an attempt to estimate the effects of a treatment when, for ethical or practical reasons, it is not possible to randomly assign units to treatment or control; see Cochran (1965) for a review of issues that arise in such studies. The central problem in observational studies is that treated and control units may not be comparable prior to treatment, so differences in outcomes in treated and control groups may or may not indicate effects actually caused by the treatment. This problem has two aspects: The treated and control groups may be seen to differ prior to treatment with respect to various recorded measurements, or they may be suspected to differ in ways that have not been recorded. Observed pretreatment differences are controlled by adjustments, for example by matched sampling, the method discussed here. Even after adjustments have been made for recorded pretreatment differences, there is always a concern that some important differences were not recorded, so no adjustments could be made. See Rosenbaum (1987a,b) and the references given there for discussion of methods for addressing unobserved pretreatment differences.

Pretreatment measurements are available for N treated units, numbered $n = 1, \ldots, N$, and a reservoir of M potential controls, numbered $m = 1, \ldots, M$. Often M is much larger than N, but this is not essential, and it is assumed only that $M \geq N$. Each unit has a vector of pretreatment measurements, say $\mathbf{x}_n$ for the nth treated unit and $\mathbf{w}_m$ for the mth potential control. A *matched pair* is an ordered pair (n, m) with $1 \leq n \leq N$ and $1 \leq m \leq M$, indicating that the nth treated unit is matched with the mth potential control. A *complete matched-pair sample* is a set $\mathcal{S}$ of N disjoint matched pairs, that is, N matched pairs in which each treated unit appears once, and each control appears either once or not at all. An *incomplete matched-pair sample* is a set of $<N$ disjoint matched pairs; however, there are strong reasons for avoiding incomplete matched-pair samples (Rosenbaum and Rubin 1985a), and little attention will be given to them here.

There are two notions of a "good" complete matched-pair sample. The first involves closely matched individual pairs, and the second involves balanced treated and control groups. A pair (n, m) is closely matched if $\mathbf{x}_n$ is in some sense close to $\mathbf{w}_m$, for instance, close in terms of some distance, $\delta(\mathbf{x}_n, \mathbf{w}_m)$. When there is only a single pretreatment measurement or covariate (i.e., when $\mathbf{x}_n$ and $\mathbf{w}_m$ are scalars), the distance typically studied is the absolute difference in their values (e.g., Rubin 1973). When there are several covariates, various distances have been used, including the Euclidean distance based on standardized coordinates and the Mahalanobis distance (Car-

* Paul R. Rosenbaum is Associate Professor, Department of Statistics, Wharton School, University of Pennsylvania, Philadelphia, PA 19104.

Journal of the American Statistical Association
December 1989, Vol. 84, No. 408, Theory and Methods

penter 1977; Cochran and Rubin 1973, sec. 6; Rubin 1980; Smith, Kark, Cassel, and Spears 1977). Another possibility involves replacing coordinates by their ranks. Alternatively, one might try to weight different coordinates by some measure of importance. (Although the distance must be nonnegative, it need not be a true distance; it need not satisfy the triangle inequality.) The total distance between matched pairs, $\sum_{(n,m)\in\mathcal{S}} \delta(\mathbf{x}_n, \mathbf{w}_m)$, is one measure of the quality of the matched sample.

When there are more than a few covariates, genuinely close matched pairs will be rare. This motivates the second notion of a good matched-pair sample, namely covariate balance. There is covariate balance if within the matched sample the distributions of $\mathbf{x}_n$ and $\mathbf{w}_m$ are similar for matched units, that is, for units with $(n, m) \in \mathcal{S}$. For instance, the vector difference $\bar{\mathbf{d}}$ in covariate means in the matched treated and control groups is one of many measures of covariate imbalance. Note that $\bar{\mathbf{d}}$ may be small (i.e., close to the $\mathbf{0}$ vector) even if some individual matched pairs exhibit large differences, $\mathbf{x}_n - \mathbf{w}_m$, because the differences in different pairs may cancel. Balance is therefore a weaker condition than close matching within each pair, and since it is weaker it can often be attained when close matching within pairs is not possible.

One way to obtain balanced matched samples is by matching on the propensity score, as discussed by Rosenbaum and Rubin (1983). Under a stochastic model for the assignment of units to treatment or control, the propensity score is the conditional probability of being assigned to treatment, given the observed covariates. In practice, the propensity score is estimated from the data using a model such as a logit model. As the propensity score is a scalar, it is often easy to obtain close matches on it, and theoretical arguments show that the resulting matched sample will tend to balance all of the covariates used to construct the propensity score; that is, the dimensionality of $\mathbf{x}_n$ and $\mathbf{w}_m$ ceases to be a major problem. The balance obtained in this way is stochastic, that is, in expectation and with probability 1 as N tends to ∞, but in any given matched sample some imbalances will remain. An empirical investigation (Rosenbaum and Rubin 1985b) compared the performance of three greedy matching methods as applied to a data set. The best of these three picked the closest match in terms of the Mahalanobis metric from a restricted subset or caliper of potential controls who were close to the treated unit on the propensity score.

Optimal matching within propensity score calipers is discussed in Sections 3.4 and 4.2; arguably, it is the method of choice.

1.3 A Motivating Example: Optimal Matching Versus a Greedy Heuristic

In this section, a greedy match is contrasted with an optimal match obtained by one of the methods discussed in later sections. The matching algorithms in the statistical literature are essentially greedy algorithms; they do not generally find a matched sample that minimizes the total distance between matched pairs. The details of the optimal method and the results obtained by other methods are deferred to later sections. The example selected was large enough to be interesting, but small enough to permit direct examination using a single table of distances. In particular, the example exhibits the bottleneck problem that optimal matching methods resolve in the best possible way.

The example uses the now-familiar data on 26 U.S. light water nuclear power plants, as collected by W. E. Mooz and as reported by Cox and Snell (1981). (Excluded are the six "partial turnkey" plants, whose costs may contain hidden subsidies.) Seven of the plants were constructed on a site at which a light water reactor had previously existed; they are the treated units. Each such unit will be matched with two controls from among the remaining 19 plants. A comparison of the costs of the treated and control plants might be the basis for thinking about the advantages or disadvantages of building a new plant at an existing site. (Of course, such an analysis would involve analytical issues beyond the construction of the matched sample discussed here; e.g., see Rosenbaum 1988a,b.)

Table 1 is a matrix of distances between treated and control power plants, with the 7 treated plants as the columns and the 19 potential controls as the rows. For easy identification, the plant numbers are those of Cox and Snell (1981), so plant 3 is the first treated plant, plant 1 is the first potential control, and the distance between these two plants is 28. The distance between two plants is defined in terms of two covariates: the date the construction permit was issued and the capacity of the power plant.

Table 1. Distances Between Treated and Control Power Plants

	Treated plants						
Control plants	*3*	*5**	*9*	*18*	*20*	*22**	*24*
1*	28	24	10	7	17	20	14
2	[0]	3	18	28	20	31	32
4*	3	[0]	14	24	16	28	29
6*	22	22	18	8	32	35	30
7	14	[10]	4	14	18	20	18
8	30	27	12	[2]	26	29	24
10*	17	14	[5]	10	20	22	17
11	28	26	11	6	18	20	16
12*	26	24	9	12	12	14	9
13	28	24	10	[0]	24	26	22
14	20	16	14	24	[0]	12	12
15	22	19	12	22	[2]	9	10
16	23	20	[5]	4	20	22	17
17*	26	23	14	24	6	[5]	6
19	21	18	22	32	7	15	16
21	[18]	16	10	20	4	12	14
23	34	31	16	18	14	9	[4]
25	40	37	22	16	20	11	[8]
26	28	25	28	38	14	[12]	17

NOTE: An optimal match is indicated by a box.
* Plants constructed in the northeastern part of the United States.

These two covariates were replaced by their ranks (1, . . . , 26), with average ranks used in case of ties. The distance between two plants is the sum of the two absolute differences in their ranks on the two covariates. A distance of 0 indicates two plants had identical tied values for both covariates, whereas the maximum possible difference is $(26 - 1) + (26 - 1) = 50$. The actual differences range from 0 to 40. (I have had some unpleasant experiences using standard deviations to scale covariates in multivariate matching, and I am inclined to think that either ranks or some more resistant measure of spread should routinely be used instead.)

A greedy algorithm divides a problem such as matching into N separate decisions, makes those decisions sequentially without revision or reconsideration, and each decision is best among the choices then available. (Stepwise regression by forward selection is a familiar example of a greedy algorithm.) The greedy algorithm starts with a match of minimum distance, in this case one of the zero distances, removing that row from further consideration. For instance, it might match treated plant 3 with potential control 2, removing plant 2 from further consideration. The process repeats on the reduced array. For instance, plant 5 might be matched with plant 4. As soon as a treated plant has two matched controls, the corresponding column is also deleted.

Table 2 contrasts the performance of the greedy and optimal matching procedures. Each step in Table 2 is the addition of one control to the matched sample, so at step k there is a partial match consisting of k controls, with at most two controls matched to each treated unit. The total distance within these k pairs is used to evaluate the match at step k. The greedy algorithm performs perfectly for the first 11 steps; no partial match with k controls is better than greedy's choice for $k \leq 11$. At step 12, greedy misses a small opportunity: It adds match $(n, m) = (22, 26)$ at a cost of 12 units of distance, whereas a new match could have been added at a cost of 11 by removing match (20, 15), thereby freeing control 15, and adding matches (20, 21) and (22, 15), for a total cost of $-2 + 4 + 9 = 11$. At step 13, greedy misses a somewhat larger opportunity, and it is now behind by a cost of 6 units of distance. At step 14, with two controls per treated unit, greedy misses another small opportunity, and has paid a total price that is $(79 - 71)/71 = 11\%$ higher than necessary.

Table 2. Comparison of Greedy and Optimal Match Construction

	Greedy		Optimal	
Step	Action	Total distance	Action	Total distance
1	Add (3, 2)	0	Add (3, 2)	0
2	Add (5, 4)	0	Add (5, 4)	0
3	Add (18, 13)	0	Add (18, 13)	0
4	Add (20, 14)	0	Add (20, 14)	0
5	Add (18, 8)	2	Add (18, 8)	2
6	Add (20, 15)	4	Add (20, 15)	4
7	Add (9, 7)	8	Add (9, 7)	8
8	Add (24, 23)	12	Add (24, 23)	12
9	Add (22, 17)	17	Add (22, 17)	17
10	Add (9, 10)	22	Add (9, 10)	22
11	Add (24, 25)	30	Add (24, 25)	30
12	Add (22, 26)	42	Delete (20, 15)	
			Add (20, 21)	
			Add (22, 15)	41
13	Add (5, 21)	58	Delete (9, 7)	
			Add (9, 16)	
			Add (5, 7)	52
14	Add (3, 19)	79	Delete (22, 15)	
			Delete (20, 21)	
			Add (22, 26)	
			Add (20, 15)	
			Add (3, 21)	71

1.4 Comparing Greedy and Optimal Matching

Why prefer the optimal match? There are several reasons. First, there is the obvious point that the optimal match is always as good as and often better than the greedy match. In the example, the loss due to greedy was 11%; not a disaster, but worth avoiding.

The second point, though distinct, is closely related to the first. Although a greedy algorithm (like forward stepwise regression) may provide a tolerable answer, it rarely comes with a guarantee that the answer is in fact tolerable. In particular, greedy matching can be arbitrarily poor compared to optimal matching. To see this, consider a case with $N = M = 2$, and the following 2×2 distance matrix.

		Treated	
		1	2
Control	1	0	ε
	2	ε	∞

For $0 < \varepsilon < \infty$, greedy grabs the (1, 1) match at a cost of 0, can never reconsider, and is forced to pay a cost of ∞ for the (2, 2) match. Of course, the optimal match is (1, 2) and (2, 1) with a cost of 2ε. There is no simple way to be sure you are not paying an intolerably high price using greedy. In short, even if the goal is a tolerable rather than an optimal match, greedy comes with no guarantee that it will find a tolerable match when it exists. Korte and Hausmann (1978) evaluated greedy heuristics for maximum similarity and minimum distance matching; surprisingly, these cases turn out to be quite different.

Consider a second larger example that permits a fairly complete evaluation. Suppose there are N treated units having covariate values $2, 4, \ldots, 2N$ and an equal number of potential controls having covariate values $1 - \varepsilon, 3 - \varepsilon, \ldots, 2N - 1 - \varepsilon$, where $\varepsilon > 0$ is vanishingly small. Suppose that the absolute difference in the covariate values is used as the measure of distance. Then, 2 is slightly closer to $3 - \varepsilon$ than to $1 - \varepsilon$, and so forth. Greedy pairs 2 with $3 - \varepsilon$, 3 with $4 - \varepsilon$, and so on, and is finally forced to pair $2N$ with the only unmatched treated unit, namely $1 - \varepsilon$. If the covariate were age, this would mean matching the oldest treated unit to the youngest control. Since ε is vanishingly small, the total absolute difference within the N pairs is $\Delta_G = (N - 1) + (2N - 1)$. In contrast, the optimal procedure pairs 2 with $1 - \varepsilon$, 3 with $2 - \varepsilon$, and so on, for a total distance of $\Delta_O = N$. The percent increase in distance due to using greedy rather than optimal matching is $100(\Delta_G - \Delta_O)/\Delta_O = 100\{2 - (2/N)\} \rightarrow 200\%$ as

$N \to \infty$, so greedy can be quite poor in large problems as well as small ones. Note, however, that if ε is negative and vanishingly small, then greedy yields the optimal match. In other words, greedy's performance relative to optimal matching is sensitive to small changes in the covariate values.

The two previous examples concern pair matching when $N = M$, so every control is matched. As a result, the marginal distribution of the covariate among the M controls is unchanged by matching. As a final example, consider pair matching with $M = N + 1$, so one control is to be left unmatched. For simplicity, consider the same situation as in the previous paragraph but with one additional control having covariate value $3N$, so this additional control is in effect an outlier in the covariate. For vanishingly small $\varepsilon > 0$, greedy pairs 2 with $3 - \varepsilon$, and so on, finally pairing $2N$ with $3N$ since $2N$ is closer to $3N$ than to $1 - \varepsilon$. So greedy unnecessarily uses the outlier, yielding a total absolute distance of $N - 1 + N = 2N - 1$. The optimal match is unchanged from the previous paragraph, with a total distance of N. Consider the treated-minus-control difference in covariate means for the N matched pairs. For the greedy match the difference in means is $-(2N - 1)/N$ whereas for the optimal match it is 1, so greedy matching increased the absolute value of the difference in covariate means by about 100%. Suppose there is an outcome Y that is related to the covariate and the treatment by a standard analysis of covariance model, that is, a model with an additive treatment effect and which is linear in the covariate. Suppose further that the adjustment for the covariate is made using matching rather than analysis of covariance, and that no supplementary adjustments for the covariate are made. Since everything is linear, the bias in Y remaining after matching for the covariate is proportional to the difference in covariate means, so in this example greedy matching increases the bias in Y by about 100% above what it would be for optimal matching. [See Cochran and Rubin (1973) for many calculations of this kind.]

The third reason for preferring an optimal match is relevant when there are calipers, such as the propensity score calipers in Section 1.2. Calipers forbid the matching of certain treated units to certain controls. When there are calipers, a complete pair matching may or may not exist. Even if a complete pair matching exists, greedy may not find it, whereas the optimal matching procedure illustrated in Section 1.3 will always find a complete pair matching if it exists, and otherwise will find an optimal matching of maximum size. A small example suffices to show that greedy may fail to find a complete pair matching when it exists. In the aforementioned 2×2 distance matrix, suppose that an infinite distance indicates a forbidden match. Then, treated unit 1 has a caliper that includes both controls, whereas treated unit 2 has a caliper that includes only control 1. For $\varepsilon > 0$, greedy pairs treated unit 1 with control 1, and then has no match for treated unit 2. The optimal matching procedure discovers a complete match with a cost of 2ε. Larger examples can obviously be constructed.

2. NETWORK FLOW THEORY: A SHORT REVIEW

2.1 Graphs, Networks, Flows, Maximum Flows, and Minimum Cost Flows

One version of optimal matching is a standard problem that is known to be equivalent to finding a flow of minimum cost in a network. Section 2.1 reviews various definitions, and Section 2.2 discusses the matching problem. Minimum cost flows have been discussed in many standard references, including Ford and Fulkerson (1962, sec. 3), Lawler (1976, sec. 4), Carré (1979, sec. 6), and Papadimitriou and Steiglitz (1982, sec. 7).

A (directed) *graph* is a set of vertices V and a set E of (directed) edges consisting of ordered pairs $e = (v_1, v_2)$ of distinct vertices, so E is a subset of $V \times V$. In the discussion here, V is not empty and contains finitely many elements. One draws a picture of a graph by drawing a point for each vertex $v \in V$ and, for each edge $e = (v_1, v_2) \in E$, an arrow from v_1 to v_2. An edge $e = (v_1, v_2)$ is said to be from v_1 to v_2, or to leave v_1 and enter v_2. Let $\iota(v)$ and $\theta(v)$ be the sets of all edges entering and leaving vertex v respectively. Here, ι is for in and θ is for out.

For us, a *network* is a graph with two distinguished vertices, a source $s \in V$ and a sink $t \in V$, with $\iota(s) = \emptyset$ and $\theta(t) = \emptyset$. The structures associated with network flow problems have acquired a metaphorical terminology suggesting a flow of material from the source to the sink along the edges; however, many if not most applications of network flow ideas have nothing to do with actually moving materials around. A *flow* f is a function that assigns to each edge $e \in E$ a nonnegative real number, $f(e)$, such that, excluding the source and the sink, the flow into each vertex equals the flow out from each vertex:

$$\sum_{e \in \iota(v)} f(e) = \sum_{e \in \theta(v)} f(e) \quad \text{for all } v \in V - \{s, t\}.$$

In the metaphor, $f(e)$ is the amount of material moved along edge e, and except for the source and the sink, the amount of material flowing into each vertex equals the amount flowing out. The *value of the flow*, written as $|f|$, is the net flow out from the source, which may be shown to equal the net flow into the sink, so a flow of value $|f|$ must satisfy the following linear equations in the $f(e)$'s:

$$\begin{aligned} \sum_{e \in \theta(v)} f(e) - \sum_{e \in \iota(v)} f(e) &= |f| && \text{for } v = s \\ &= 0 && \text{for all } v \in V - \{s, t\} \\ &= -|f| && \text{for } v = t. \end{aligned}$$

Each edge e has a nonnegative *capacity* $c(e)$, which may be $+\infty$. A flow is *feasible* if $0 \le f(e) \le c(e)$ for each $e \in E$. In other words, a feasible flow is an assignment of numbers to edges that satisfies certain linear equalities and inequalities. As this might suggest, several standard network problems, including optimal matching, may be solved as linear programming problems, although specialized algorithms for networks may be faster and more economical in their use of storage.

A flow is *integral* if the flow across each edge is an

integer, that is, if $f(e)$ is an integer for each e. In the metaphor, an integral flow is one that moves "whole units." A flow f is a *maximum flow* if f is feasible and if there is no other feasible flow f' such that $|f'| > |f|$. Note that there may be more than one maximum flow. A network may have bottlenecks, so the maximum flow may (and usually does) have flow values that are less than the capacities; that is, a maximum flow f will often have $f(e) < c(e)$ for some $e \in E$. It may be shown that if all of the capacities are integers, then there is a maximum flow that is integral; that is, we need not split up whole units to move the maximum possible amount of flow. As will be seen, in statistical matching problems all edge capacities are integers and only integral flows correspond to matched samples.

Each edge $e = (v_1, v_2)$ also has a nonnegative *cost* $q(e)$, with $0 \le q(e) \le \infty$, the cost of shipping one unit of material from v_1 to v_2 along e. The *total cost* of a flow f is $Q(f) = \sum_{e \in E} f(e)\, q(e)$. A flow f is a *minimum cost flow* if it is feasible and every other feasible flow with the same flow value has a total cost at least as high as the cost of f; that is, if f' is feasible and $|f| = |f'|$ then $Q(f) \le Q(f')$. In the metaphorical language of network flow theory, if f is a minimum cost flow and you want to ship $|f|$ units of material from the source to the sink, you could not do it more cheaply than by shipping $f(e)$ units along edge e, for each $e \in E$. There may be more than one flow of minimum cost. Again, it may be shown that if all of the edge capacities (capacities, *not* costs) are integers, and a flow f exists with integer flow value $|f|$, then there exists a minimum cost flow g with value $|f|$ that is integral; that is, we can ship $|f|$ units of flow at the least possible cost without dividing whole units. Section 4 discusses computing a minimum cost flow.

2.2 A Standard Matching Problem Viewed as a Minimum Cost Flow

One matched sampling problem, that of determining a complete matched-pair sample to minimize a total distance $\sum_{(n,m)\in\Im} \delta(\mathbf{x}_n, \mathbf{w}_m)$, is a standard optimization problem, called the "personnel assignment" problem (Kuhn 1955), which is well known to be equivalent to a certain minimum cost flow problem. Figure 1 displays the relevant network. The vertex set V consists of the N treated units and the M potential controls, plus a source and a sink. There is an edge e from the source to each treated unit; it has capacity $c(e) = 1$ and cost $q(e) = 0$. There is an edge from each treated unit (say n) to each potential control (say m); it has capacity $c(e) = 1$ and cost $q(e) = \delta(\mathbf{x}_n, \mathbf{w}_m)$. Finally, there is an edge e from each potential control to the sink; it has capacity $c(e) = 1$ and cost $q(e) = 0$. Since $M \ge N$, the maximum value of a flow is N. Since all of the capacities are integers, there is at least one integral maximum flow, that is, an integral flow of value N. For every maximum integral flow in this network, the flow from the source to each treated unit is 1, the flow out from each treated unit is 1 along exactly one edge and 0 along all other edges, and finally N of the M potential controls have a unit flow to the sink. In other words, every maximum integral flow determines a matching of the N treated units to N distinct controls. Again, since the capacities are integers, there is a minimum cost flow that is integral; it is the complete pair matching $\Im$ that minimizes the total distance $\sum_{(n,m)\in\Im} \delta(\mathbf{x}_n, \mathbf{w}_m)$.

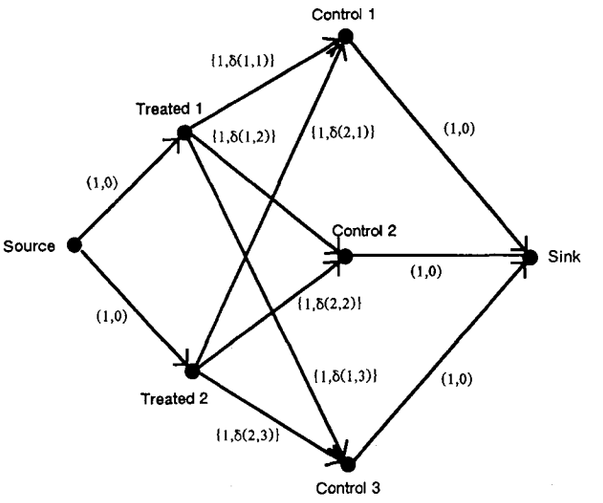

Figure 1. A Small Network for Minimum Distance Pair Matching. Each edge e is labeled by (capacity, cost) = (c(e), q(e)). The cost connecting a treated subject to a potential control is the distance between them.

3. OPTIMAL MATCHING IN OBSERVATIONAL STUDIES

3.1 Matching With Multiple Controls

The standard method of Section 2.2 can be used for minimum distance pair matching. Minimum distance matching with multiple controls is also easy. If $2N \le M$, a matched sample with two controls for each treated unit is a set $\Im$ of $2N$ pairs (n, m), with $1 \le n \le N$ and $1 \le m \le M$ such that each treated unit appears in two pairs and each potential control appears in at most one pair. (In data analysis, these $2N$ pairs are treated as N matched triples, but the set notation is convenient for constructing matched samples.) To match two controls to each treated unit when $2N \le M$, simply increase from 1 to 2 the capacity of each edge from the source to a treated unit, leaving all other edges unchanged. A maximum flow now has value $2N$, and an integral maximum flow connects each treated unit with two distinct controls. An integral flow of value $2N$ that minimizes the cost is a matched sample of minimum distance. This is the approach illustrated in Section 1.3.

3.2 Balanced Matching

So far, the only criterion used in optimal matching has been the total distance. As an alternative, it is possible to find the matching that minimizes the total distance among matchings that balance a discrete variable. For instance, in the optimal matching in Table 2, 29% = 2/7 of the treated plants were constructed in the northeastern part of the United States, whereas only 21% = 3/14 of the

control plants were constructed there. A matched sample with two controls would be balanced if 29% = 4/14 of the controls had been constructed in the Northeast. A balanced match must include among the controls an additional plant constructed in the Northeast, bringing the total from 3 to 4. The goal is to find the best balanced matched sample, that is, the minimum distance balanced matched sample.

Suppose that units fall into k mutually exclusive categories; here, $k = 2$. Balance means the treated and control units have the same distribution across the categories, that is, the same proportion in category 1, and so on. Modify the network in Figure 1 as follows. Remove the edges connecting the controls to the sink. Add k new vertices, one for each category. Place an edge of capacity 1 and cost 0 from each control to the category to which that control belongs. Place an edge from each category to the sink with cost 0 and capacity equal to the number of controls to be selected from this category for the matched sample. In the example in Table 1, there are two new vertices, *Northeast* and *Other*, and the edge (Northeast, Sink) has capacity 4, whereas the edge (Other, Sink) has capacity 10 = (14 − 4). An integral flow of value 14 must draw 4 controls from the Northeast and 10 from Other regions, and so must be a balanced matched sample. In this network, an integral flow of minimum cost is a balanced match that minimizes the total distance among all balanced matched samples. So the problem is again one of finding a minimum cost integral flow in a specific network.

The optimal balanced matched sample includes the following triples, with the treated unit first and an asterisk indicating a plant constructed in the Northeast: (3, 2, 10*), (5*, 4*, 7), (9, 12*, 16), (18, 8, 13), (20, 14, 21), (22*, 15, 17*), (24, 23, 25). The optimal balanced match has a total distance of 74.5, which is only 5% greater than the optimal match with a total distance of 71. In contrast, the greedy match in Table 2 is unbalanced and has a total distance of 79, which is 6% higher than the optimal balanced match. In short, the optimal balanced match is better than the greedy match in two ways: It is balanced and has a smaller total distance.

3.3 Matching With a Variable Number of Controls

When matching is used as a tool in data analysis rather than as a basis for selecting a sample of controls, it will often be advantageous to retain all M controls. If M is a multiple of N, the method of Section 3.1 could be used to assign M/N controls to each treated unit. However, this would not work in the example of Section 1.3, where $N = 7$ and $M = 19$, for in this case some treated units must have at least three controls, whereas others have no more than two. There is, then, the additional question: How many controls should be assigned each treated unit? Which units should get more and which less?

Clearly, each treated unit should have at least one control, or there would be no basis for comparison. Fix two integers (α, β) with $\alpha \leq \beta$, such that $\alpha \geq 1$ is the minimum number of controls permitted for a treated unit, and $\beta \leq M - N + 1$ is the maximum number permitted. A matched sample $\mathfrak{I}$ is a set of M pairs (n, m), with $1 \leq n \leq N$ and $1 \leq m \leq M$, such that each n appears in at least α and at most β pairs, and each m appears in exactly one pair. Subject to the (α, β) restrictions, a matching $\mathfrak{I}$ is found to minimize the total distance between matched treated and control units.

The network for matching with a variable number of controls is given in Figure 2. There is an edge from the source to each treated unit with capacity α. These N edges will carry the flow for α controls matched to each treated unit, that is, a total flow of αN will cross these N edges. There is a new mode called *extras*. The edge from the source to extras has capacity $M - \alpha N$, the total number of extra matches to be allocated beyond the minimum of α. If the total flow into the sink is to be M so that all controls are matched, then M units of flow must leave source, which implies that all of the edges leaving the source are at full capacity. In particular, this means that at least α units of flow must enter each treated unit, so each treated unit must have at least α matched controls. There is an edge from extras to each treated unit, with capacity $\beta - \alpha \geq 0$. Flow can enter a treated unit only through the two edges that enter it, which have capacities α and $\beta - \alpha$, so the total flow into a treated unit is at most β, and at most β controls are matched to a treated unit. All $M - \alpha N$ units of flow into extras must exit into the treated units. A minimum cost integral flow of value M is then a matching that satisfies the restrictions on the number of controls for each treated unit and that minimizes the total distance subject to these restrictions.

Table 3 continues the example of Section 1.3. Table 3 shows three matched samples obtained for three choices of α and β. The first of the three ($\alpha = 1, \beta = 13$) allows any number of controls, providing only that each treated unit gets at least one. In fact, under the optimal matching,

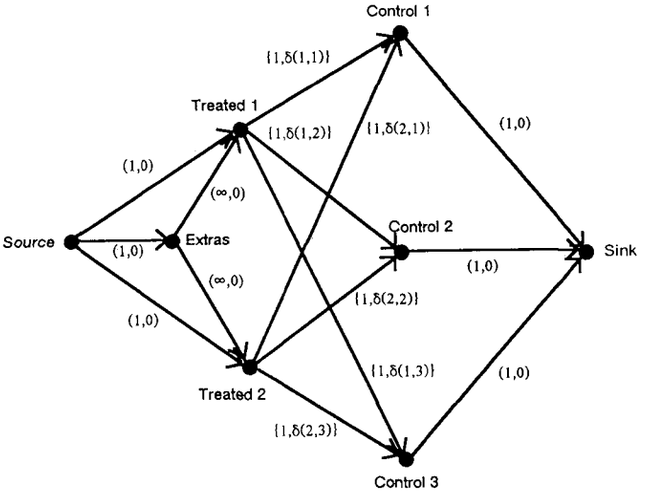

Figure 2. Matching With a Variable Number of Controls. Extra controls, that is, controls beyond the first control assigned to each treated subject, flow through the node "Extras."

Table 3. Optimal Matching With a Variable Number of Controls

	Matched controls		
Treated unit	$\alpha = 1, \beta = 13$	$\alpha = 1, \beta = 4$	$\alpha = 2, \beta = 3$
3	2	2	2, 6
5	4	4	4, 7
9	7, 10, 12	1, 7, 10, 16	1, 10, 16
18	1, 6, 8, 11, 13, 16	6, 8, 11, 13	8, 11, 13
20	14, 15, 19, 21	14, 15, 19, 21	14, 19, 21
22	17, 26	17, 26	15, 17, 26
24	23, 25	12, 23, 25	12, 23, 25
$\sum_{(n,m)\in\mathfrak{I}} \delta(\mathbf{x}_n, \mathbf{w}_m)$	90	94	121

treated unit 18 had six matched controls, whereas units 3 and 5 had only one. The total distance was 90, and no matching of all 7 treated units with all 19 controls can give a smaller total distance. The second matched sample in Table 3 ($\alpha = 1, \beta = 4$) required no more than four controls per treated unit. It had a total distance of 94, only slightly higher than the first matching. The third matched sample in Table 3 ($\alpha = 2, \beta = 3$) paid a high price to obtain two or three controls for each treated unit: The total distance increased to 121. In particular, the match (3, 6) contributed a distance of 20, and the match (5, 7) contributed a distance of 10.5.

3.4 Optimal Matching With Propensity Score Calipers

The optimal matching methods in Sections 2.2 and 3.1–3.3 may all be combined with the use of propensity score calipers. The only change is that the edge set E no longer includes all pairs of treated and control subjects, and instead is restricted to pairs that are close on the propensity score. As noted in Section 1.2, propensity score calipers can increase covariate balance, and in this sense are very desirable. As noted in Section 4, propensity score calipers can increase the speed of the algorithms that find the optimal match, so in this sense too, the calipers are desirable. The calipers reduce the number of edges in the network, which can substantially increase the speed of calculation.

4. COMPUTING MINIMUM COST FLOWS

4.1 Algorithms for Minimum Cost Flow

There is an elegant, extensively studied theory for finding minimum cost flows. This theory has two intertwined aspects: certain mathematical results and efficient algorithms. In this respect, the problem of finding a minimum cost flow resembles the problem of calculating a least squares fit: There is a mathematical theory that pretends arithmetic with real numbers is possible, and a computational theory that is greatly concerned with round-off error. In network flow problems, the computational issue is not round-off error, but rather the number of steps required to produce a solution.

The mathematical results, of which there are several, say that if you can improve a given flow, then the improvement has a simple form. Two such approaches are briefly described in the following. The algorithms for minimum cost flow look for these simple improvements, and quit when none can be found. The mathematical theory says this strategy works. For an attractive discussion of these formal results, see Carré (1979, sec. 6).

Both of the approaches to be described use a displacement or residual network that indicates both unused flow capacity and the possibility of rechanneling existing flow. Here, unused flow capacity means the ability to match a previously unmatched unit, whereas rechanneling existing flow means the ability to unmatch two previously matched units and match them to different units. The residual network is defined in relation to a given flow f, and it has the same vertex set V as the original network. The residual network has two types of edges, normal and inverted. A normal edge is an edge of the original network, $e \in E$. The capacity of a normal edge is $c(e) - f(e)$, since $f(e)$ units of flow are already moving across e, so that at most $c(e) - f(e)$ units of flow can be added. The cost for a normal edge in the residual network is the cost $q(e)$ in the original network; that is, moving another unit of flow across the e will cost $q(e)$. In addition to the normal edges, the displacement network had inverted edges; that is, for each $e = (v_1, v_2) \in E$, there is an inverted edge $e^* = (v_2, v_1)$ going back from v_2 to v_1. The capacity of the inverted edge e^* is the flow across e, namely $f(e)$. The cost of shipping a unit of flow on an inverted edge e^* is the negative of the cost of the normal edge, namely $-q(e)$. This says that we could, if we wished, shut down $f(e)$ units of flow across e, which is the same as shipping $f(e)$ units of flow back along e^*. (The description just given is adequate here because all of our original networks are free of cycles, so the original network cannot include both e and e^*. More complicated networks require a slightly different discussion.)

Table 2 illustrates one method of finding a minimum cost flow, namely *minimum cost augmentation*. The method starts with the flow of value zero, $f(e) = 0$ for all $e \in E$. It then finds the least costly path from the source to the sink, and pushes one unit of flow along that path. At this first step, the path connects the source to the treated unit with the smallest distance to a control, then connects that treated unit to this closest control, and finally connects the control to the sink. Each edge e on this path now has $f(e) = 1$, and the total flow value $|f|$ is 1 (i.e., there is one matched pair). The residual network is then determined, the minimum cost path of nonzero capacity from its source to its sink is found, and a unit of flow is pushed along this path. This increases the number of matched pairs by 1. The process repeats, always increasing the flow by one unit along a path of minimum cost in the residual network. This path may take the form (source, treated unit i, control j, treated unit k, control h, sink), which gets rid of the old pair (k, j), replacing it with two new pairs, (i, j) and (k, h). This happened several times in Table 2. Here, (k, j) is an inverted edge in the residual network, since $(j, k) \in E$. It is possible to prove that at each step in this process, we have a minimum cost integral flow of value $|f|$, that is, a minimum distance matching with $|f|$ matched pairs (e.g., Carré 1979, sec. 6.6; Tarjan 1983, sec. 8.4).

The method just described starts with the zero flow and optimally increases the flow by one unit; that is, it starts with no matched pairs and adds them one at a time. An alternative method, *cost reducing cycles*, starts with a flow of the desired value $|f|$ and reduces its cost; that is, it starts with a matched sample of the desired size and reduces its total distance. For instance, we might start with a greedy match and improve it. To find improvements, we look at the residual network. The method of cost reducing cycles looks for negative cost cycles, uses them to reduce the total cost or distance, and quits when no more cycles exist. See Carré (1979, sec. 6.6) or Tarjan (1983, sec. 8.4) for detailed discussion.

The computational theory is concerned with finding these simple improvements in an efficient way. To indicate roughly what is involved, a few examples follow. For instance, it is easier to find a minimum cost augmenting path if all costs are positive (using Dijkstra's algorithm) than if some are negative, so one computational device tranforms the displacement network to eliminate negative costs (Edmonds and Karp 1972). Also, finding a minimum cost augmenting path involves repeatedly finding the smallest element in a gradually changing set. If the n elements of the set are stored haphazardly in an array or a linked list, it takes n steps to find the minimum. An alternative is to store the elements in a heap ordered tree so that the smallest element is always handy; see Fredman and Tarjan (1987) for a recent approach and references to earlier approaches. Two attractive discussions of the computational theory were given by Tarjan (1983, sec. 8.4) and Mehlhorn (1984, sec. 9.3).

4.2 Efficiency of Minimum Cost Flow Algorithms for Large Matching Problems

The usual way to study the efficiency of algorithms is in terms of the number of operations required to handle large problems. Specifically, if T is a measure of the size of a matching problem and a particular algorithm requires up to aT^3 operations for some $a > 0$, then $O(T^3)$ operations are required. A fast sorting algorithm can sort T numbers into order from smallest to largest in $O\{T\log(T)\}$ operations. Multiplying two $T \times T$ matrices takes $O\{T^3\}$ operations if performed in the conventional way and $O\{T^{2.81}\}$ operations if performed using Strassen's method. For motivation and discussion of this standard approach to computational efficiency, see Aho, Hopcroft, and Ullman (1974), Knuth (1973), or Wilf (1986).

The networks in Section 3 are among the simplest types of networks: They have integer capacities, no cycles, and nonnegative costs. Within this class of networks, it is useful to make several distinctions. By the definition of an edge, the number of edges $|E|$ in a network is at most $|V|^2$. In the matching problem, $|V|$ measures the number of units; that is, $|V| = O(T)$, where $T = N + M$. As we consider a sequence of larger networks—that is, as we allow $|V| \to \infty$ and $|E| \to \infty$—several things can happen. If there is a constant c $(0 < c < \infty)$ such that $|V|^2/|E| \to c$, then the network is said to be *dense*, whereas if $|V|/|E| \to c$, the network is *sparse*. A network is *intermediate* if $\lim|V|/|E| = 0$ and $\lim|V|^2/|E| = \infty$. The degree of density of the sequence of networks determines the complexity of the network algorithm, so we need to consider how dense and sparse networks arise.

Assume that as the number of units to be matched increases, the relative size of the treated and control groups is unchanged; that is, $M/N \to$ constant as $T = N + M \to \infty$. Then, the networks in Sections 3.1–3.3 are all dense; that is, $|V| = O(T)$ and $|E| = O(T^2)$, because every treated unit is connected with every control, so there are more than $NM = O(T^2)$ edges. On the other hand, a sparse network may be obtained using propensity score calipers in the following way. Fix some integer $K \geq 2$, perhaps $K = 10$. The caliper for treated unit n is the set of K controls that have the closest values of the propensity score. Note that calipers for distinct treated units will often overlap. In the network, an edge links treated unit n and control m if and only if control m is in the caliper for treated unit n. The number of edges connecting treated and control units is now KN, and $|E| = O(T)$, so the network is sparse. If the caliper size K is not fixed, but rather is allowed to grow slowly with T, then intermediate networks may be obtained. Recall from Section 1.2 that the use of propensity score calipers can improve distributional balance in matching.

The minimum cost matching in a dense network may be found in $O(T^3)$ operations, which is the same order as for multiplying two $T \times T$ matrices by the conventional method. In a sparse network, the matching may be found in $O\{T^2\log(T)\}$ operations, which is the same order as sorting T unrelated arrays of size T, and is of a smaller order than matrix multiplication. Derivations of these bounds may be found in Papadimitriou and Steiglitz (1982, sec. 11.2), Tarjan (1983, sec. 8.4), and Mehlhorn (1984, sec. 9.3). In all cases, including intermediate networks, a bound of $O\{T^2\log(T) + T|E|\}$ is attainable using the new algorithm of Fredman and Tarjan (1987), where as always $|E| \leq T^2$. Note that for intermediate networks the new Fredman and Tarjan algorithm is an improvement on the $O(T^3)$ bound for dense networks, since $|E|$ is small compared to T^2 in intermediate networks.

With calipers, there may be no matching that includes all N treated units, because of the nature of the overlap between the calipers. The method of minimum cost augmentation will find an optimal match of the largest possible size, including as many treated units as possible.

There are several ways to implement optimal matching, including the following.

1. Arguably the best general approach currently available uses minimum cost augmentation with the Edmonds and Karp (1972) transformation of the residual network, together with some version of Dijkstra's algorithm. See Tarjan (1983, sec. 8.4) and Mehlhorn (1984, sec. 9.3) for detailed discussion of this approach, and Fredman and Tarjan (1987) for a method that attains the best current time bound for dense, sparse, and intermediate networks.
2. For matched pairs, the labeling implementation of

the Hungarian method may be used. It attains the $O(T^3)$ time bound for a dense network and is easy to implement. See Papadimitriou and Steiglitz (1982, sec. 11.2) for details.

3. Someone who very much wanted to use a widely available software package could use linear programming methods directly, say using the IMSL subroutine for the simplex algorithm. This may be slow and inefficient in its use of space. Aspects of this approach were discussed by Lawler (1976) and Papadimitriou and Steiglitz (1982).

[*Received July 1988. Revised March 1989.*]

REFERENCES

Aho, A., Hopcroft, J., and Ullman, J. (1974), *The Design and Analysis of Computer Algorithms*, Reading, MA: Addison-Wesley.

Carpenter, R. (1977), "Matching When Covariables Are Normally Distributed," *Biometrika*, 64, 299–307.

Carré, B. (1979), *Graphs and Networks*, New York: Oxford University Press.

Cochran, W. G. (1965), "The Planning of Observational Studies of Human Populations" (with discussion), *Journal of the Royal Statistical Society*, Ser. A, 128, 134–155.

Cochran, W. G., and Rubin, D. B. (1973), "Controlling Bias in Observational Studies: A Review," *Sankhyā*, Ser. A, 35, 417–446.

Cox, D. R., and Snell, E. J. (1981), *Applied Statistics: Principles and Examples*, London: Chapman & Hall.

Edmonds, J., and Karp, R. (1972), "Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems," *Journal of the Association of Computing Machinery*, 19, 248–264.

Ford, L., and Fulkerson, D. (1962), *Flows in Networks*, Princeton, NJ: Princeton University Press.

Fredman, M., and Tarjan, R. (1987), "Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms," *Journal of the Association of Computing Machinery*, 34, 596–615.

Knuth, D. (1973), *The Art of Computer Programming*, Reading, MA: Addison-Wesley.

Korte, B., and Hausmann, D. (1978), "An Analysis of the Greedy Heuristic for Independence Systems," *Annals of Discrete Mathematics*, 2, 65–74.

Kuhn, H. (1955), "The Hungarian Method for the Assignment Problem," *Naval Research Logistics Quarterly*, 2, 83–98.

Lawler, E. (1976), *Combinatorial Optimization: Networks and Matroids*, New York: Holt, Rinehart & Winston.

Mehlhorn, K. (1984), *Data Structures and Algorithms 2: Graph Algorithms and NP-Completeness*, New York: Springer-Verlag (translation of the 1977 German edition).

Papadimitriou, C., and Steiglitz, K. (1982), *Combinatorial Optimization: Algorithms and Complexity*, Englewood Cliffs, NJ: Prentice-Hall.

Rosenbaum, P. R. (1987a), "Sensitivity Analysis for Certain Permutation Tests in Matched Observational Studies," *Biometrika*, 74, 13–26; Correction (1988), 75, 396.

——— (1987b), "The Role of a Second Control Group in an Observational Study" (with discussion), *Statistical Science*, 2, 292–316.

——— (1988a), "Permutation Tests for Matched Pairs With Adjustments for Covariates," *Applied Statistics*, 37, 401–411.

——— (1988b), "Sensitivity Analysis for Matching With Multiple Controls," *Biometrika*, 75, 577–581.

Rosenbaum, P., and Rubin, D. (1983), "The Central Role of the Propensity Score in Observational Studies for Causal Effects," *Biometrika*, 70, 41–55.

——— (1985a), "The Bias Due to Incomplete Matching," *Biometrics*, 41, 106–116.

——— (1985b), "Constructing a Control Group Using Multivariate Matched Sampling Methods That Incorporate the Propensity Score," *The American Statistician*, 39, 33–38.

Rubin, D. (1973), "Matching to Remove Bias in Observational Studies," *Biometrics*, 29, 159–183; Correction (1974), 30, 728.

——— (1980), "Bias Reduction Using Mahalanobis Metric Matching," *Biometrics*, 36, 293–298.

Smith, A., Kark, J., Cassel, J., and Spears, G. (1977), "Analysis of Prospective Epidemiologic Studies by Minimum Distance Case-Control Matching," *American Journal of Epidemiology*, 105, 567–574.

Tarjan, R. (1983), *Data Structures and Network Algorithms*, Philadelphia: Society for Industrial and Applied Mathematics.

Wilf, H. (1986), *Algorithms and Complexity*, Englewood Cliffs, NJ: Prentice-Hall.