PSYCHOMETRIKA—VOL. 78, NO. 3, 498–525
JULY 2013
DOI: 10.1007/S11336-012-9313-1

MULTIOBJECTIVE BLOCKMODELING FOR SOCIAL NETWORK ANALYSIS

MICHAEL BRUSCO

COLLEGE OF BUSINESS, FLORIDA STATE UNIVERSITY

PATRICK DOREIAN

DEPARTMENT OF SOCIOLOGY, UNIVERSITY OF PITTSBURGH

FACULTY OF SOCIAL SCIENCES, UNIVERSITY OF LJUBLJANA

DOUGLAS STEINLEY

DEPARTMENT OF PSYCHOLOGICAL SCIENCES, UNIVERSITY OF MISSOURI, COLUMBIA

CINTHIA B. SATORNINO

COLLEGE OF BUSINESS, FLORIDA STATE UNIVERSITY

To date, most methods for direct blockmodeling of social network data have focused on the optimization of a single objective function. However, there are a variety of social network applications where it is advantageous to consider two or more objectives simultaneously. These applications can broadly be placed into two categories: (1) simultaneous optimization of multiple criteria for fitting a blockmodel based on a single network matrix and (2) simultaneous optimization of multiple criteria for fitting a blockmodel based on two or more network matrices, where the matrices being fit can take the form of multiple indicators for an underlying relationship, or multiple matrices for a set of objects measured at two or more different points in time. A multiobjective tabu search procedure is proposed for estimating the set of Pareto efficient blockmodels. This procedure is used in three examples that demonstrate possible applications of the multiobjective blockmodeling paradigm.

Key words: social networks, blockmodeling, multiobjective programming, heuristics, tabu search.

1. Introduction

1.1. *Blockmodeling and Structural Equivalence*

One early goal of social psychologists and sociologists was the understanding of role relations in social groups. This required collecting and organizing social relational data to reveal the fundamental network structure of social groups. Blockmodeling was a tool originally designed to accomplish this task. Although its origin stems from the study of role relations, blockmodeling has also become a useful tool for discerning the fundamental structure of *any* social network.

The key idea behind blockmodeling is to think of (social) actors as being equivalent in terms of the distribution of their ties in their social group. Equivalent actors are grouped into clusters, called *positions*, and this imposes also a partition of the relational ties into blocks. A *block* is composed of the set of ties between the actors in a pair of positions. If there are n actors in a group, then the size of its network is $(n \times n)$, and if there are K positions, then the blockmodel is a $(K \times K)$ network (with $K \ll n$). The simpler and much smaller $(K \times K)$ network is an *image*

Requests for reprints should be sent to Michael Brusco, College of Business, Florida State University, Tallahassee, FL 32306-1110, USA. E-mail: mbrusco@fsu.edu

of the larger ($n \times n$) network. This image captures the underlying (essential) structure of the network, one that is more easily understood than the original network. In brief, blockmodeling methods partition the nodes and ties of a network in an attempt to reveal the basic structural properties of that network.

A key to successful blockmodeling is defining equivalence among actors. Lorrain and White (1971) introduced "structural equivalence" where actors are equivalent if they are connected to exactly the same other actors in the network: They are structurally identical and can be put in the same cluster. The premise of structural equivalence informed attempts to create programs that establish blockmodel images of networks, while recognizing that most networks do not conform exactly to structural equivalence. Broadly, these programs can be classified into two categories: Indirect and direct approaches. Breiger, Boorman, and Arabie (1975) and Burt (1976) offered widely used algorithms for blockmodeling that are indirect in the sense that they transform a network into a set of (dis)similarity measures among actors and then cluster them. The problem with such indirect approaches is that there are many (dis)similarity measures and many potential clustering algorithms. Arbitrary choices about these items affect the blockmodels identified. Batagelj, Ferligoj, and Doreian (1992) noted that if a network does conform to structural equivalence, then the only possible types of blocks are null blocks (containing only null ties) and complete blocks (having no null ties). This permits a direct approach to fitting blockmodels by minimizing an explicit criterion function operationalizing departures from exact structural equivalence. Doreian, Batagelj, and Ferligoj (1994) found that the direct algorithm of Batagelj et al. (1992) frequently outperformed the Breiger et al. (1975) algorithm in terms of fit and never performed worse in terms of fitting the data.

Doreian, Batagelj, and Ferligoj (2005) present generalized blockmodeling as a general approach to fitting blockmodels to social network data where a much wider set of ideal block types, beyond those implied by structural equivalence, are mobilized. Examples include ranked clusters models for classroom networks and nobility marriage systems, column-regular and row-regular block types for primate grooming networks, row-dominant and column-dominant blocks for Little League baseball teams and student discussion groups, and signed networks (with positive and negative blocks) for study of organizational data. Depending on the substantive context, a wide range of block types and blockmodel forms can be fitted to social networks. Nevertheless, structural equivalence remains the most used equivalence type in blockmodeling. Žnidaršič, Ferligoj, and Doreian (2012) show that structural equivalence is quite robust, even in the presence of substantial measurement error, in returning the correct structure of social networks in blockmodel form. Given its wide use and robustness, we focus on structural equivalence here.

1.2. Blockmodeling Variants

In addition to defining equivalence types, other important issues relevant for blockmodeling merit attention. Brusco and Steinley (2011) noted that blockmodeling can be categorized along a number of dimensions: (1) deterministic vs. stochastic, (2) one-mode vs. two-mode networks, (3) signed vs. unsigned networks, and (4) exploratory vs. confirmatory blockmodeling. Perhaps the most critical dimension pertains to the absence or presence of the assumption of a probabilistic model. Deterministic blockmodeling methods (Batagelj et al., 1992; Brusco, Doreian, Mrvar, & Steinley, 2011; Brusco & Steinley, 2007, 2009a, 2011; Doreian et al., 2005; Doreian & Mrvar, 1996; Hartigan, 1972) do not assume a probabilistic model and typically rely on exact or approximate algorithms to find blockmodels that minimize some measure of inconsistency with an ideal block structure. Contrastingly, stochastic blockmodeling methods (Airoldi, Blei, Fienberg, & Xing, 2008; Bickel & Chen, 2009; Fienberg, Meyer, & Wasserman, 1985; Goldenberg, Zheng, Fienberg, & Airoldi, 2009; Govaert & Nadif, 2003; Handcock, Raftery, & Tantrum, 2007; Holland, Laskey, & Leinhardt, 1983; Nowicki & Snijders, 2001; Steinley, Brusco, & Wasserman, 2011; Wasserman & Faust, 1994) posit an underlying statistical model and incorporate maximum likelihood or pseudo-maximum likelihood methods to obtain blockmodels.

One-mode blockmodeling methods produce a partition of a single set of objects that define the nodes of the network, whereas two-mode methods obtain two separate but simultaneous partitions of the two sets of nodes. Unsigned networks assume that edges of the network are either absent or positively weighted when present, but signed networks allow negative weights as well (e.g., enemy ties as well as friendly ties that can vary also in strength). Exploratory blockmodeling methods make no assumption about the placement of block types within a blockmodel, whereas confirmatory methods do.

1.3. Multiobjective Blockmodeling

In general, direct blockmodeling provides a general framework for fitting blockmodels to social relational data by using an explicit criterion function capturing the difference between an ideal blockmodel structure and an empirical network. However, despite this generality, there remain problems. Even for a single network, it is possible that multiple equally well-fitting blockmodels (having a minimized criterion function) exist. Although the blockmodel structure may be stable, this presents serious problems with regard to interpreting such blockmodels. Moreover, when partitioning networks, using one criterion function seems limited. A more general approach would enable simultaneous optimization of multiple criteria. This is explicitly tackled here as a general problem that can pervade many aspects of blockmodeling.

We consider blockmodeling as a multiobjective combinatorial optimization problem. There are a number of advantages and opportunities associated with adopting this framework. One is that it can mitigate the well-known problem of multiple equally well-fitting partitions that often arises in blockmodeling (Doreian et al., 2005; Doreian & Mrvar, 2009). However, as more objective criterion functions are considered, the potential for equally well-fitting partitions diminishes. In this context, equally well-fitting partitions for one criterion are mitigated by the fact that other criteria are, effectively, breaking the ties among those partitions.

A second advantage of multiobjective blockmodeling is that it can preclude the need to specify weights for violations from an ideal block structure. For example, Doreian et al. (2005), Brusco and Steinley (2009a, 2010), and others have discussed the use of differential weights for ones in null blocks and zeros in complete blocks. This is, effectively, an implicit transformation or weighted-sum approach to a multiobjective problem that considers these two types of violations independently. However, if we can approximate the Pareto efficient set for such problems, then a judicious selection can be made from the frontier, obviating the need for an a priori specification of weights.

A third opportunity for multiobjective programming arises in situations where an analyst has multiple network matrices for the same set of objects. This can occur when measurements are taken at multiple points in time; examples include Sampson's (1968) monastery data and Newcomb's (1961) fraternity data. They also can arise from multiple indicators of the same relation (e.g., Lemann & Solomon, 1952) or multiple relations for a set of actors (e.g., Sampson, 1968). Multiobjective programming can be used to establish a blockmodel that fits multiple matrices well. To illustrate, assume that the optimal blockmodel for network matrix one fits network matrix two rather poorly, and vice versa. In many instances, multiobjective programming can identify a blockmodel that fits both network matrices (one and two) reasonably well.

We use B to represent a blockmodel for network data and $f_g(B)$ as the measure of fit of the blockmodel for criterion g for each of G criteria ($1 \leq g \leq G$). The relevant criteria differ depending on the application and may include the number of inconsistencies from an ideal block structure, the correlation of the block with an ideal pattern, or a more complex (e.g., likelihood-based) measure. The focal point of multiobjective blockmodeling is the identification of *Pareto efficient* (or *nondominated*) blockmodels. To clarify this concept, we define $\mathcal{B}$ as the set of all blockmodels[1] and let B correspond to a blockmodel from $\mathcal{B}$ (i.e., $B \in \mathcal{B}$). We assume, without loss of generality, that each of the G objectives should be minimized. In addition, a blockmodel B' *dominates* B if $f_g(B') \leq f_g(B)$ for all $1 \leq g \leq G$ and $f_g(B') < f_g(B)$ for one or more of the G objectives. The Pareto efficient (or nondominated) set, $\mathcal{B}^{\mathrm{ND}} \subset \mathcal{B}$, consists of all blockmodels, B', that are not dominated by any other blockmodel of the same type.

One possible approach for approximating[2] the set of nondominated blockmodels in $\mathcal{B}^{\mathrm{ND}}$ is to transform the multiobjective problem into a single-objective function that consists of a weighted sum of the objectives (Brusco & Stahl, 2001; Ehrgott & Wiecek, 2005; Ferligoj & Batagelj, 1992). This approach to multiobjective programming is hereafter referred to as the weighted-sum approach. Among the limitations of this approach are: (1) the challenge of choosing an appropriate set of weights and (2) the failure to identify unsupported Pareto efficient solutions.[3] The a priori selection of an appropriate set of weights is difficult because the analyst typically does not know precisely how the multiple objective functions will behave for different weights. Moreover, even if a broad set of weights is evaluated, there are often large gaps in the estimated nondominated set where interesting unsupported Pareto efficient solutions have been missed.

Given these shortcomings of the weighted-sum approach, we adopt what is known as a direct procedure for estimating the nondominated set of blockmodels. More specifically, we obtain results using an adaptation of the multiobjective tabu search (MOTS) paradigm developed by Kulturel-Konak, Smith, and Norman (2006) and also discussed by Kulturel-Konak, Coit, and Baheranwala (2008). The multiobjective tabu search algorithm attempts to generate the entire nondominated set of blockmodels (both supported and unsupported). Our adaptation of the MOTS paradigm contains some critical modifications that substantially improve the quality of the estimated Pareto efficient set. The most important of these modifications are: (1) creating an initial population of the estimated Pareto efficient set with (near-)optimal blockmodels for each of the single objectives independently, (2) using finer granularities achieved by random selection of alternative weighting schemes for the objective functions rather than the all-or-nothing random selection of one criterion and the exclusion of all others, and (3) having a more exhaustive search prior to the switching of weighting schemes. Finer granularity allows for the search of more neighborhoods of the solution space, whereas an exhaustive search enables thoroughness of the investigation of each neighborhood.

In the next section, we describe the general purpose multiobjective tabu search algorithm that can be used for multiobjective blockmodeling. This is followed by Section 3, which focuses on an application within the context of generalized structural balance blockmodeling of a signed network. Section 4 reports the results for an application related to fitting a blockmodel based on multiple relations for the same set of objects. Section 5 offers an application to a larger two-mode unsigned network. Results for two sets of simulated data are provided in Section 6, along with precise recommendations for multiobjective blockmodeling. The paper concludes in Section 7 with a brief summary, as well as discussion of limitations and extensions.

2. Multiobjective Tabu Search Algorithm

Section 2.1 provides a brief description of tabu search, with particular emphasis on multiobjective tabu search. Section 2.2 provides a glossary of terms used for the presentation of the

[1] Typically, we are referring to the set of all blockmodels for a specific number of clusters or positions.

[2] We use the term "approximating" the nondominated set here because a guarantee that a blockmodel is nondominated can only be verified for small networks where exact solutions are feasible. More commonly, heuristics will be used for larger networks to obtain an estimation or approximation of the nondominated set without affording such a guarantee.

[3] The weighted-sum approach yields "supported" Pareto efficient solutions; however, there can also be "unsupported" Pareto efficient solutions that cannot be identified using any set of weights (see Ulungu & Teghem, 1994, for a discussion).

algorithm. The steps of the algorithm are presented in Section 2.3, and selection of a solution from the nondominated set is discussed in Section 2.4.

2.1. *Overview of Tabu Search*

Originally developed by Glover (1989, 1990), tabu search is a popular metaheuristic for discrete optimization problems (see Glover & Laguna, 1993 for a survey). The "tabu" in tabu search involves the prohibition of specific neighborhood search operations (e.g., a relocation of an object to a different cluster) for a user-specified number of iterations, which is often referred to as the tabu length. By disallowing some neighborhood search moves, the tabu search algorithm is able to diversify the search and avoid returning to the same local optimum. This diversification often enables the algorithm to locate better local optima. Nevertheless, tabu search is a heuristic method, and, although a global optimum might be obtained, the algorithm is not guaranteed to find one.

In their adaptation of tabu search for multiobjective combinatorial optimization, Kulturel-Konak et al. (2006) randomly select one of the G objective functions to optimize at any particular time. Standard local-search operations such as pairwise interchange or object relocation are used to find the best non-tabu solution. Intensification of the search occurs as improved solutions are obtained by these operations. Diversification of the search can occur in a couple of different ways. First, a new objective function to optimize is randomly selected after a user-specified number of iterations have elapsed. In addition, after a prescribed number of iterations with no change in the nondominated set, the search process is restarted after replacing the current incumbent solution with one randomly selected from the nondominated set. A precise description of our implementation for multiobjective blockmodeling is provided in the following subsections.

2.2. *Glossary*

$\mathcal{B}$ the set of all blockmodels;

$\mathcal{B}^{\mathrm{ND}}$ the set of nondominated blockmodels;

$\mathcal{B}^{\mathrm{tabu}}$ the set of blockmodels on the tabu list;

B^{I} the incumbent blockmodel in the algorithm;

n the number of objects in the network;

$\mathbf{w}$ a set of objective function weights $[w_1, \ldots, w_G]$, where $w_g \geq 0$ is the weight for criterion g for $1 \leq g \leq G$ and $\sum_{g=1}^{G} w_g = 1$;

$\mathbf{W}$ a $q \times G$ matrix of alternative weighting schemes, where each row contains a set of objective function weights, $\mathbf{w}$. The number of weighting schemes, q, is an input parameter selected by the user;

$F(\mathbf{w}, B)$ the weighted objective function value, $F(\mathbf{w}, B) = \sum_{g=1}^{G} w_g f_g(B)$, for blockmodel B given weighting scheme $\mathbf{w}$;

$\psi_{\max}$ the termination criterion, which is defined as the maximum number of iterations permitted with no modification of the nondominated solutions list, $\mathcal{B}^{\mathrm{ND}}$; when this limit is reached, the algorithm terminates;

$\xi_{\max}$ the diversification limit, which is defined as a limit on number of iterations with no modification of the nondominated solutions list since the last diversification; when the limit is reached, diversification occurs via selection of a new incumbent solution from $\mathcal{B}^{\mathrm{ND}}$;

$\tau_{\max}$ the maximum length of the tabu list, $\mathcal{B}^{\mathrm{tabu}}$;

ψ a counter for the number of iterations with no modification in $\mathcal{B}^{\mathrm{ND}}$;

ξ a counter for the number of iterations with no modification in $\mathcal{B}^{\mathrm{ND}}$, since the last diversification;

τ a counter for the number of iterations at the current tabu length.

2.3. The Algorithm

Step 0. *Initialization*. Randomly construct an incumbent blockmodel, B^{I}, and place it in the nondominated set of blockmodels such that $\mathcal{B}^{\mathrm{ND}} = \{B^{\mathrm{I}}\}$. Alternatively, $\mathcal{B}^{\mathrm{ND}}$ can be initially populated by applying exact or approximation procedures to the single-objective optimization problems associated with each of the G criteria individually. Initialize parameters of the tabu search process: $\psi_{\max} = 1000$; $\xi_{\max} = 10$; $\tau_{\max} = \mathit{random\_integer}[n, 3n]$; $\psi = 0$, and $\xi = 0$.

Step 1. *Select weighting scheme*. Randomly select a weighting scheme, $\mathbf{w}$ from $\mathbf{W}$.

Step 2. *Increment counters*. Set $\tau = \tau + 1$, $\xi = \xi + 1$, and $\psi = \psi + 1$.

Step 3. *Dynamic adjustment of tabu list length*. If $\tau = \tau_{\max}$, then reset $\tau_{\max} = \mathit{random\_integer}[n, 3n]$ and $\tau = 0$.

Step 4. *Neighborhood search*. Conduct a search of the neighborhood of the incumbent blockmodel, B^{I}, consisting of all single-object relocations. For each trial blockmodel, B, that is generated, perform the following substeps:

Step 4a. Add B to $\mathcal{B}^{\mathrm{ND}}$ if B is nondominated.
Step 4b. Eliminate all blockmodels from $\mathcal{B}^{\mathrm{ND}}$ that are dominated by B.
Step 4c. If $\mathcal{B}^{\mathrm{ND}}$ is modified in Step 4a or 4b, then set $\xi = 0$ and $\psi = 0$.

Step 5. *Update tabu list*. Choose the blockmodel $B \notin \mathcal{B}^{\mathrm{tabu}}$ in Step 4 that minimizes $F(\mathbf{w}, B)$. Append B to the bottom of the tabu list, $\mathcal{B}^{\mathrm{tabu}}$. If $|\mathcal{B}^{\mathrm{tabu}}| = \tau_{\max}$, then remove the oldest blockmodel from $\mathcal{B}^{\mathrm{tabu}}$.

Step 6. *Intensification*. If $F(\mathbf{w}, B) < F(\mathbf{w}, B^{\mathrm{I}})$, then set $B^{\mathrm{I}} = B$ and go to Step 4; otherwise, go to Step 7.

Step 7. *Termination*. If $\psi = \psi_{\max}$, then return $\mathcal{B}^{\mathrm{ND}}$ and STOP; otherwise, proceed to Step 8.

Step 8. *Diversification*. If $\xi < \xi_{\max}$, then return to Step 1 with B^{I} as the current incumbent; otherwise, randomly select a new B^{I} from $\mathcal{B}^{\mathrm{ND}}$, set $\xi = 0$, and return to Step 1.

The multiobjective tabu search algorithm for blockmodeling begins in Step 0 with the random construction of an initial blockmodel, B^{I}, which is placed in the nondominated set, $\mathcal{B}^{\mathrm{ND}} = \{B^{\mathrm{I}}\}$. As an alternative, $\mathcal{B}^{\mathrm{ND}}$ can be initially populated by applying exact or approximate solution procedures to each of the G single-optimization problems independently. We evaluate the merits of this latter approach in our computational studies in Sections 6.1 and 6.2.

The parameters of the algorithm are also initialized in Step 0. We use the same $\psi_{\max} = 1000$ and $\tau_{\max} = \mathit{random\_integer}[n, 3n]$ parameter settings used by Kulturel-Konak et al. (2006) in their study. We also initially adopted the recommendation of Kulturel-Konak et al. (2006) by choosing $\xi_{\max} = 250$. However, because the relocation heuristic embedded in Step 4 of our algorithm runs until convergence to a local minimum, we have found that it is more effective to change the weighting scheme much more frequently and have opted for $\xi_{\max} = 10$. Our experience is that these settings work well across a broad range of problem sizes. A weighting scheme is selected in Step 1. This step affords an important modification from the original MOTS paradigm (Kulturel-Konak et al., 2006), which selects an objective criterion in Step 1 rather than a weighting scheme. For example, in the case of $G = 2$ criteria, the original MOTS paradigm only considers switching the selected criterion from $f_1(B)$ to $f_2(B)$ and vice versa. In our framework of weighting schemes, this corresponds to two choices at Step 1: ($w_1 = 1$ and $w_2 = 0$) or ($w_1 = 0$ and $w_2 = 1$). One problem with this approach is that the weight of zero implies that the algorithm is indifferent to values of the secondary objective function. We recommend defining a small positive constant, ε, that will allow at least some weight on the secondary criterion so as to break ties among alternative optima. The weights would then become ($w_1 = 1 - \varepsilon$ and $w_2 = \varepsilon$) or ($w_1 = \varepsilon$ and $w_2 = 1 - \varepsilon$). The value of ε should be chosen sufficiently small so that no sacrifice in the criterion weighted $1 - \varepsilon$ would ever be permitted for an improvement in the criterion weighted ε.

In our applications, $\varepsilon = 0.001$ always suffices. Our implementation also enables the analyst to include other schemes with partial weights on both criteria, such as ($w_1 = 0.6$ and $w_2 = 0.4$) or ($w_1 = 0.3$ and $w_2 = 0.7$). The key advantage of this expansion is that it allows for a more rigorous search in more areas of the frontier of nondominated solutions.

Step 2 increments the key counters in the algorithm, namely the tabu counter (τ), the termination criterion counter (ψ), and the diversification counter (ξ). As recommended by Kulturel-Konak et al. (2006), dynamic adjustment of the length of the tabu list occurs in Step 3 every $\tau_{\max}$ iterations. The neighborhood search is conducted in Step 4 of the algorithm. Trial blockmodels are generated by attempting to relocate each object from its current cluster to one of the other clusters.[4] For each trial blockmodel, B, generated by this procedure, a check is made to see if the nondominated list, $\mathcal{B}^{\mathrm{ND}}$, should be updated. If B is not dominated by any blockmodel in the current list, then it is appended to the list in Step 4a. If any blockmodels in $\mathcal{B}^{\mathrm{ND}}$ are dominated by the newly added blockmodel in Step 4a, then those dominated blockmodels are removed from the list in Step 4b. If $\mathcal{B}^{\mathrm{ND}}$ is modified in Step 4a (and, possibly, Step 4b as well), then the two counters for the number of iterations without modification of $\mathcal{B}^{\mathrm{ND}}$ are reset to $\xi = 0$ and $\psi = 0$.

The tabu list is updated in Step 5 by appending the non-tabu blockmodel from the neighborhood search that yields the best weighted objective function value. If the updated tabu list exceeds the maximum list size (i.e., if $|\mathcal{B}^{\mathrm{tabu}}| = \tau_{\max}$), then the first (oldest) blockmodel in the list is deleted. If the weighted objective function value for the best trial blockmodel is an improvement over the incumbent (i.e., $F(\mathbf{w}, B) < F(\mathbf{w}, B^{\mathrm{I}})$), then the search process for the current weighting scheme is intensified at Step 6 by setting $B^{\mathrm{I}} = B$ and returning to Step 4. Preserving the current weighting scheme as long as better $F(\mathbf{w}, B)$ values continue to be found represents another novel feature of our implementation of the MOTS paradigm If no better $F(\mathbf{w}, B)$ value is found in Step 6, then control is passed to Step 7 for a check of the termination condition.

The algorithm terminates in Step 7 if the number of iterations with no modification of the nondominated set reaches its maximum limit (i.e., $\psi = \psi_{\max}$), and $\mathcal{B}^{\mathrm{ND}}$ is returned as the output. After every $\xi_{\max}$ iterations with no modification of $\mathcal{B}^{\mathrm{ND}}$, diversification occurs in Step 8 by randomly selecting a new incumbent solution from the nondominated set. After the first $\psi = \xi = 10$ iterations with no modification of $\mathcal{B}^{\mathrm{ND}}$, the termination limit is not reached. Therefore, control passes from Step 7 to Step 8, where diversification occurs by randomly selecting a new incumbent solution from $\mathcal{B}^{\mathrm{ND}}$ and resetting $\xi = 0$. If other 10 iterations with no modification occur, then $\psi = 20$ and $\xi = 10$. The termination limit is still not reached, but it is again time for diversification in Step 8 and resetting of $\xi = 0$. In summary, the modification processes for ψ and ξ are as follows: (1) if a new nondominated solution is found, then both counters are reset as $\xi = 0$ and $\psi = 0$, (2) if the diversification limit is reached but the termination limit is not, then the diversification counter is reset to $\xi = 0$, but the termination counter ($\psi < 1000$) is not adjusted, or (3) if the termination limit is reached, then the algorithm terminates and $\mathcal{B}^{\mathrm{ND}}$ is returned as the output.

Multiobjective tabu search algorithms for blockmodeling have been prepared for the applications considered in this paper.[5] All programs are written in Fortran 90 and are available from the website: http://mailer.fsu.edu/~mbrusco.

[4] Only object relocations or transfers from one cluster to another are used in the neighborhood search. Pairwise interchanges were also considered, but they significantly increased computation time without any improvement in solution quality.

[5] All of the results reported herein were obtained via application of these programs on a 2.2 GHz, Pentium 4 PC with 1 GB of RAM, which is a somewhat dated (by about 10 years) hardware platform. Accordingly, the reported computation times are conservative relative to current computer capabilities.

2.4. *Evaluation of the Approximated Nondominated Set*

Ehrgott and Gandibleaux (2000) distinguish among a priori, a posteriori, and interactive multiobjective combinatorial optimization. In the a priori approach, the analyst (or decision-maker) prespecifies preferences or weights for the objective functions in advance, and these can be directly incorporated in the solution method (e.g., using the weighted-sum approach). This a priori specification is often impractical in exploratory blockmodeling because the analyst typically does not know the behavior of the individual objective functions beforehand. The a posteriori approach generates the approximated nondominated set and, subsequently, requires use of the analyst's weights and preferences to choose one or more blockmodels from this set. The interactive approach is a multistep process that requires significant analyst participation to evaluate the quality of solutions during the multiobjective programming process. We adopt a perspective similar to one outlined by Ehrgott and Gandibleaux (2000, p. 448), which blends the a posteriori and interactive perspectives. The multiobjective blockmodeling algorithm is first used to establish a good approximation of the nondominated set. Subsequently, the preferences of the analyst would be used to focus the examination on specific regions of the approximated nondominated set by re-running the algorithm with an expanded set of weighting schemes. Although some criterion measures can be established to facilitate the selection of solutions from the nondominated set, the analyst's ability to provide practical interpretation would remain an important component of real applications.

The output of the multiobjective tabu search algorithm is the approximated nondominated set, $\mathcal{B}^{\mathrm{ND}}$. The size of this set can vary widely depending on the application; however, it generally tends to increase as a function of network size and the number of objective criteria (G). The evaluation of the nondominated set can range from the selection and interpretation of a single blockmodel from the set, to an evaluation of multiple blockmodels from various regions of the set. In the case of all minimization objectives, one simple criterion is to select the solution that is the minimum distance from the origin. Unfortunately, this approach does not take into account differences in measurement scales and deviation from the best possible values for the various objective functions. As an alternative, we adopt an approach proposed by Brusco and Stahl (2001) within the more general context of multiobjective combinatorial data analysis. This approach begins with the identification of the best-found (minimum) objective function value, f_g^*, for each criterion $1 \leq g \leq G$. These numbers are then used to establish an ideal point $(f_1^*, \ldots, f_G^*)$, from which the Euclidean distance from each blockmodel in $\mathcal{B}^{\mathrm{ND}}$ can be measured to that point. An analyst can select the blockmodel from $\mathcal{B}^{\mathrm{ND}}$ based on its proximity to the ideal point or select the blockmodel with the smallest maximum distance across the G criteria.[6]

3. Example 1. Sampson's (1968) Monastery Data

3.1. *The Network Matrix*

For our first example of multiobjective blockmodeling, we use network data collected by Sampson (1968) corresponding to the relationships among trainee monks at a New England monastery. Blockmodel analyses of Sampson's (1968) monastery data have been conducted several times in the network literature (Breiger et al., 1975; Brusco & Steinley, 2010; Doreian & Mrvar, 1996; White, Boorman, & Breiger, 1976). In our example, we consider the signed network matrix of affect ties among $n = 18$ monks at time period T2. Each monk reported his three most-liked and three most-disliked monks and, accordingly, the elements of the

[6] In applications where the criteria are measured on profoundly different scales, then some normalization of the distances is advisable.

signed network matrix, $\mathbf{A} = [a_{ij}]$, range from -3 (most disliked) to $+3$ (most-liked). A value of $a_{ij} = 0$ indicates that monk i did not identify monk j as being either liked or disliked.

3.2. *Structural Balance*

We adopt the structural balance paradigm, based on Heider's (1946) structural balance theory with important refinements by Cartwright and Harary (1956), Davis (1967), and Doreian and Mrvar (1996), to analyze the monastery data. It posits that, ideally, the monks would be partitioned into distinct clusters of monks who have positive or neutral affect with other monks in their cluster, and either negative or neutral affect with monks outside their cluster. In other words, positive ties between monks should occur only within clusters and negative ties between monks occur only between clusters. As perfect structural balance seldom holds, Doreian and Mrvar offered an optimization approach to return partitions as close to those of exact structural balance as possible. Specifically, their approach identifies blockmodels by establishing a K-cluster partition of the monks that minimizes a weighted function of the $G = 2$ types of inconsistencies: (a) negative ties within clusters and (b) positive ties between clusters. Denoting $B = \{S_1, \ldots, S_K\}$ as a blockmodel with S_k representing the monks assigned to cluster k, these two objective criteria can be stated as follows:

$$f_1(B) = \sum_{k=1}^{K} \sum_{i=S_k} \sum_{j \in S_k} \max\{0, -a_{ij}\}, \tag{1}$$

$$f_2(B) = \sum_{k=1}^{K} \sum_{l \neq k} \sum_{i \in S_k} \sum_{j \in S_l} \max\{0, a_{ij}\}. \tag{2}$$

By defining α $(0 \leq \alpha \leq 1)$ as the weight assigned to objective criterion $f_1(B)$ and $(1 - \alpha)$ as the weight assigned to criterion $f_2(B)$, the single-objective, weighted-sum optimization problem studied by Doreian and Mrvar (1996) is:

$$\text{Minimize:} \quad F(\alpha, B) = \alpha f_1(B) + (1 - \alpha) f_2(B), \tag{3}$$

$$\text{subject to} \quad B \in \boldsymbol{\mathcal{B}}. \tag{4}$$

Doreian and Mrvar (1996) used a relocation heuristic for obtaining solutions to the optimization problem posed by Equations (3) and (4). Brusco and Steinley (2010) recently presented an exact algorithm that can often obtain globally optimal solutions for signed networks with 40 or fewer objects. As for other weighted-sum approaches to multiobjective optimization, a key limitation of these algorithms is that they require specification of α. Although $\alpha = 0.5$ is a common assumption, this need not be the most appropriate alternative in all applications. For example, it might be much more disadvantageous from the standpoint of group cohesion to have members that dislike each other in the same cluster, than it is to have two members who like each other be in different clusters.[7] Thus, an important benefit of our proposed multiobjective method is that it avoids the prespecification of α. Moreover, it can uncover nondominated solutions that might not be recoverable using any value of α.

3.3. *Results*

Previous results for Sampson's monastery data (see Doreian et al., 2005 and Brusco & Steinley, 2010) have shown that, when minimizing the total number of inconsistencies, the optimal number of clusters for this network matrix is $K = 3$. We used the branch-and-bound algorithm

[7] For networks with unequal proportions of positive and negative ties, having $\alpha = 0.5$ is problematic for exploratory blockmodeling and brings in the issue of needing α to depart from 0.5.

TABLE 1.
Results for Example 1—Sampson's (1968) monastery data. The objective function values for the approximation of the Pareto efficient (nondominated) set of blockmodels. The results for the unsupported Pareto efficient blockmodels are highlighted in bold.

Blockmodel #	$f_1(B)$	$f_2(B)$	Distance from ideal	Cluster assignments
1	0	44	30.00	{2, 7, 8, 9, 12, 16} {1, 3, 13, 17, 18} {4, 5, 6, 10, 11, 14, 15}
2	1	38	24.02	{1, 2, 7, 8, 9, 12, 15, 16} {3, 13, 17, 18} {4, 5, 6, 10, 11, 14}
3	2	33	19.10	{1, 2, 7, 12, 14, 15, 16} {3, 13, 17, 18} {4, 5, 6, 8, 9, 10, 11}
4	5	30	16.76	{1, 2, 3, 7, 12, 14, 15, 16} {13, 17, 18} {4, 5, 6, 8, 9, 10, 11}
5	**8**	**29**	**17.00**	**{1, 2, 5, 7, 8, 9, 11, 12, 14, 15, 16} {3, 13, 17, 18} {4, 6, 10}**
6	9	28	16.64	{1, 2, 3, 7, 12, 14, 15, 16, 18} {13, 17} {4, 5, 6, 8, 9, 10, 11}
7	**13**	**27**	**18.38**	**{1, 2, 5, 7, 8, 9, 11, 12, 14, 15, 16, 18} {3, 13, 17} {4, 6, 10}**
8	16	25	19.42	{1, 2, 3, 7, 12, 14, 15, 16, 17, 18} {13} {4, 5, 6, 8, 9, 10, 11}
9	**21**	**24**	**23.26**	**{1, 2, 4, 5, 7, 8, 9, 10, 11, 12, 14, 15, 16} {3, 13, 17, 18} {6}**
10	**30**	**22**	**31.05**	**{1, 2, 5, 7, 8, 9, 11, 12, 13, 14, 15, 16, 18} {3, 17} {4, 6, 10}**
11	31	20	31.58	{1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 16} {3, 17} {13, 18}
12	42	17	42.11	{1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 16, 18} {3, 17} {13}
13	60	14	60.00	{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 16, 18} {13} {17}

developed by Brusco and Steinley (2010) to initially populate the approximated nondominated set with optimal blockmodels for the single-objective optimization problems associated with Equations (1) and (2). The computation times to obtain these solutions were 0.03 and 0.02 seconds, respectively. Next, the multiobjective tabu search heuristic was applied to the network matrix for $K = 3$ using two different weighting schemes, one coarse the other fine. The coarse weighting scheme was $\mathbf{W} = [(0.999, 0.001), (0.001, 0.999)]$, whereas the fine weighting scheme was $\mathbf{W} = [(0.999, 0.001); (0.9, 0.1); (0.8, 0.2); (0.7, 0.3), \ldots, (0.3, 0.7), (0.2, 0.8), (0.1, 0.9), (0.001, 0.999)]$. Both weighting schemes and parameters yielded the same nondominated set of 13 distinct blockmodels, shown in Table 1 with their objective function values and cluster assignments.[8] A visual display of the Pareto efficient frontier is provided in Figure 1.

The solid lines on the Pareto efficient frontier in Figure 1 clearly distinguish the supported Pareto efficient blockmodels from those that are unsupported: the supported blockmodels are those connected by the line, whereas the unsupported blockmodels fall above the line. To illustrate the distinction, consider blockmodels #4, #5, and #6 from Table 1, which are displayed as the fourth square marker, first circular marker, and fifth square marker from left to right in Figure 1. Supported blockmodels #4 and #6 are connected by a solid line, but the unsupported blockmodel #5 lies just above that line. Assuming $\alpha = 1/3$, the weighted sum for blockmodel #4 is $(1/3)(5) + (2/3)(30) = 65/3$, which is equal to the weighted sum for blockmodel #6 at $(1/3)(9) + (2/3)(28) = 65/3$. The weighted sum for blockmodel #5 is slightly greater at $(1/3)(8) + (2/3)(29) = 66/3$. Therefore, blockmodel #4 has the smallest weighted sum over the range $0 \leq \alpha < 1/3$, blockmodels #4 and #6 have equal weighted sums at $\alpha = 1/3$, and blockmodel #6 has the smallest weighted sum over the range $1/3 < \alpha \leq 1$. There is no value of α such that blockmodel #5 has the smallest weighted sum, and thus it is not *supported* by any weighting scheme.

[8] The computation times for the multiobjective tabu search algorithm using the coarse and fine weighting schemes were 0.17 and 0.14 seconds, respectively. Thus, for this relatively small network matrix with $G = 2$, the performance of the algorithm is not sensitive to the weighting scheme. We also ran the multiobjective tabu search algorithm using the same two weighting schemes, but this time only initially populating the nondominated set with a random starting solution. The computation times were 0.25 and 0.19 seconds for the coarse and fine weighting schemes, respectively, and the final nondominated sets were the same as those obtained when using the branch-and-bound algorithm for the initial population of the nondominated set.

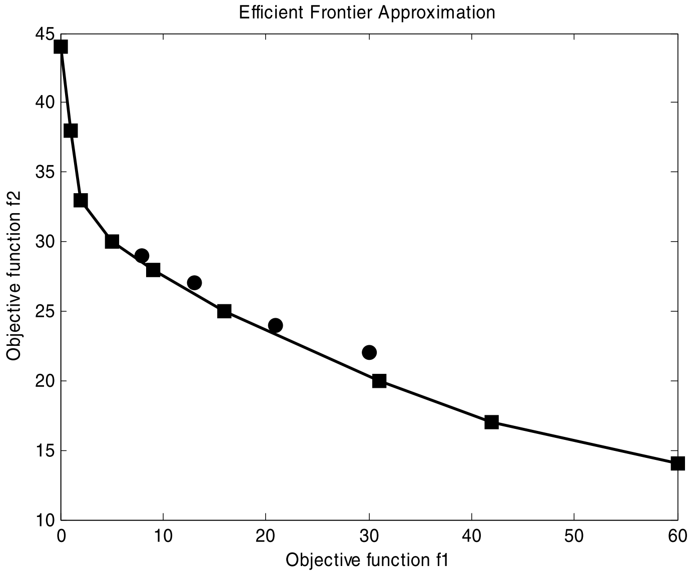

FIGURE 1.
The approximation of the Pareto efficient (nondominated) set for Sampson's (1968) monastery affect data. The *solid line* connects the *nine square markers* that correspond to supported Pareto efficient blockmodels, whereas the *four circular markers above the line* correspond to the unsupported Pareto efficient blockmodels.

Blockmodel #3 is precisely the same as the one reported by Doreian et al. (2005, p. 319), which minimizes the total number of inconsistencies at $f_1(B) + f_2(B)$ (i.e., $2 + 33) = 35$. Blockmodel #4 (obtained by Breiger et al., 1975) also has 35 consistencies, and, accordingly, blockmodels #3 and #4 would be considered as equally well-fitting partitions at $K = 3$ for a single-objective optimization problem focusing on minimizing the total sum of inconsistencies. These two blockmodels are displayed in Table 2. It is important to recognize, however, that these two blockmodels are only equally well-fitting under the assumption that the two types of inconsistencies are equally important (e.g., assuming $\alpha = 0.5$ in the weighted-sum context). For $\alpha < 0.5$, blockmodel #4 is preferred,[9] whereas blockmodel #3 would be superior for $\alpha > 0.5$.

The Euclidean distance of each blockmodel from the ideal point also can be used by the analyst as part of the comparative process. The ideal point for the network matrix is ($f_1^* = 0$, $f_2^* = 14$). Based on the Euclidean distances of the objective functions for each blockmodel to this point, blockmodel #6 is preferred with a distance of 16.64. Blockmodel #4 is a close second with a distance of 16.76 and might be favored over blockmodel #6 if there was a desire for a smaller value of $f_1(B)$.

The analysis of the monastery network data demonstrates the principal advantage of the multiobjective tabu search algorithm, a full picture of the efficient frontier. This full picture enables the analyst to make an informed selection of one or more blockmodels.

[9] The cluster in this partition with three actors was labeled as "the Outcasts," consistent with Sampson's (1968) narrative. A closer reading of his ethnographic narrative suggests that Amand belonged with the Outcasts which is the partition reported by Doreian et al. (2005). This suggests a way of incorporating information from different sources or of different types via such weighting schemes.

TABLE 2.

Blockmodels #3 (top panel) and #4 (bottom panel) from Table 1 for Sampson's (1968) monastery data. The $K = 3$ boxes along the main diagonal of each matrix encapsulate the within-cluster elements. Inconsistencies (highlighted in bold) are the negative elements within the clusters and the positive elements between the clusters.

	1	2	7	12	14	15	16	3	13	17	18	4	5	6	8	9	10	11
1 John	0	0	**−1**	0	1	0	0	**2**	0	0	0	0	**3**	−2	0	0	−3	0
2 Gregory	3	0	2	0	1	0	0	0	−3	−2	0	0	0	0	0	0	−1	0
7 Mark	0	2	0	0	0	0	3	0	0	0	0	−3	−1	−2	**1**	0	0	0
12 Winifrid	3	2	0	0	1	0	0	−1	0	0	0	−3	0	0	0	0	0	−2
14 Hugh	3	0	0	2	0	2	0	0	−3	−1	0	0	0	0	−2	0	0	**1**
15 Boniface	3	2	0	0	1	0	0	−2	−3	−1	−1	0	0	0	0	0	0	0
16 Albert	1	2	3	0	0	0	0	0	−1	−3	−2	0	0	0	0	0	0	0
3 Basil	**2**	**3**	0	0	0	0	0	0	0	1	0	−1	0	0	−3	−2	0	0
13 Amand	0	−3	**1**	−1	0	0	0	0	0	0	3	0	**2**	−2	0	0	0	0
17 Elias	0	0	0	0	0	0	0	3	2	0	1	−3	−2	0	0	0	0	−1
18 Simplici	**2**	**3**	**1**	0	0	0	−1	0	0	0	0	−3	0	−2	0	0	0	0
4 Peter	0	0	−3	0	0	0	0	−2	0	0	−1	0	3	1	0	0	2	0
5 Bonavent	0	0	0	0	0	0	0	0	**1**	0	0	3	0	0	0	0	0	2
6 Berthold	**1**	0	−3	−2	0	0	0	0	0	0	0	3	0	0	**−1**	2	0	0
8 Victor	**3**	**2**	0	0	−2	0	0	−3	0	−1	0	0	0	0	0	1	0	0
9 Ambrose	0	0	0	0	0	0	**1**	−3	0	−2	−1	0	2	0	3	0	0	0
10 Romuald	0	0	0	0	**2**	0	0	0	0	0	0	3	0	0	1	0	0	0
11 Louis	0	0	0	0	**2**	0	0	−1	−3	−2	0	0	3	0	1	0	0	0

	1	2	3	7	12	14	15	16	13	17	18	4	5	6	8	9	10	11
1 John	0	0	2	**−1**	0	1	0	0	0	0	0	0	**3**	−2	0	0	−3	0
2 Gregory	3	0	0	2	0	1	0	0	−3	−2	0	0	0	0	0	0	−1	0
3 Basil	2	3	0	0	0	0	0	0	0	**1**	0	−1	0	0	−3	−2	0	0
7 Mark	0	2	0	0	0	0	0	3	0	0	0	−3	−1	−2	**1**	0	0	0
12 Winifrid	3	2	**−1**	0	0	1	0	0	0	0	0	−3	0	0	0	0	0	−2
14 Hugh	3	0	0	0	2	0	2	0	−3	−1	0	0	0	0	−2	0	0	**1**
15 Boniface	3	2	**−2**	0	0	1	0	0	−3	−1	−1	0	0	0	0	0	0	0
16 Albert	1	2	0	3	0	0	0	0	−1	−3	−2	0	0	0	0	0	0	0
13 Amand	0	−3	0	**1**	−1	0	0	0	0	0	3	0	**2**	−2	0	0	0	0
17 Elias	0	0	**3**	0	0	0	0	0	2	0	1	−3	−2	0	0	0	0	−1
18 Simplici	**2**	**3**	0	**1**	0	0	0	−1	0	0	0	−3	0	−2	0	0	0	0
4 Peter	0	0	−2	−3	0	0	0	0	0	0	−1	0	3	1	0	0	2	0
5 Bonavent	0	0	0	0	0	0	0	0	**1**	0	0	3	0	0	0	0	0	2
6 Berthold	**1**	0	0	−3	−2	0	0	0	0	0	0	3	0	0	**−1**	2	0	0
8 Victor	**3**	**2**	−3	0	0	−2	0	0	0	−1	0	0	0	0	0	1	0	0
9 Ambrose	0	0	−3	0	0	0	0	**1**	0	−2	−1	0	2	0	3	0	0	0
10 Romuald	0	0	0	0	0	**2**	0	0	0	0	0	3	0	0	1	0	0	0
11 Louis	0	0	−1	0	0	**2**	0	0	−3	−2	0	0	3	0	1	0	0	0

4. Example 2. House Data

4.1. *The Network Matrix*

For our second example, we demonstrate multiobjective blockmodeling for finding a single blockmodel based on multiple network matrices for the same set of objects. For this example, we use data obtained by Lemann and Solomon (1952) that correspond to the measurement of four different types of network relations among female students at three different off-campus dormitories (*houses*) at an eastern college. The four relations pertained to

(1) double-dating behavior, (2) maintaining friendship after college, (3) establishing a roommate relationship, and (4) setting up a weekend visit with family. To obtain the measurements for each relation, each woman was asked to identify the three women in the house with whom they would like to engage in the activity, as well as the three women with whom they would not like to engage in the activity. This data collection process results in three sets of matrices (one for each house), with each set containing four matrices (one for each relation). Following the convention of previous blockmodeling analyses of these data (Brusco et al., 2011; Doreian, 2008), we label the sets as House A ($n = 21$), House B ($n = 17$), and House C ($n = 20$). For each set, we use network matrices ($\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \mathbf{A}_4$) measured for the $G = 4$ different relations. A matrix element $a_{ijg} = 1$ if woman i listed woman j as one of the three women with whom they would like to do activity g, -1 if woman i listed woman j as one of the three women with whom they would not like to do activity g, and 0 otherwise.

4.2. Structural Balance Across Multiple Matrices

Once again, we adopt structural balance for this application; however, there are two salient differences from the example in Section 3. First, rather than measuring multiple fit measures for a single network matrix (as in Section 3), here we use the same fit measure for multiple network matrices (i.e., matrices measured for the double-dating, friendship, roommate, and weekend relations). Second, in this example, there are $G = 4$ objective functions instead of two as in the example in Section 3. The four objective functions can be succinctly represented as follows:

$$f_g(B) = \sum_{k=1}^{K} \sum_{i \in S_k} \sum_{j \in S_k} \max\{0, -a_{ijg}\} + \sum_{k=1}^{K} \sum_{l \neq k} \sum_{i \in S_k} \sum_{j \in S_l} \max\{0, a_{ijg}\} \quad \forall 1 \le g \le G. \tag{5}$$

Thus, the objective function for each network matrix is the number of inconsistencies associated with the blockmodel, which includes both negative elements within the clusters and positive elements between the clusters. The single-objective, weighted-sum optimization problem is:

$$\text{Minimize:} \quad F(\mathbf{w}, B) = \sum_{g=1}^{4} w_g f_g(B), \tag{6}$$

$$\text{subject to} \quad B \in \mathcal{B}. \tag{7}$$

4.3. Results

Previous results for Lemann and Solomon's (1952) dormitory data have shown that a reasonable choice for the number of clusters is $K = 4$. We began by using branch-and-bound to obtain, for each of the three houses, the single-objective optimal blockmodels for $f_1(B)$, $f_2(B)$, $f_3(B)$, and $f_4(B)$ independently. The four single-objective blockmodels for each house were used to initially populate the nondominated set for that house. The multiobjective tabu search heuristic was then applied to the set of network matrices for each house, assuming $K = 4$ and using two different weighting schemes. Defining $\varepsilon = 0.001$ and $\theta = \varepsilon/(G - 1)$, the coarse weighting scheme was $\mathbf{W} = [(1 - \varepsilon, \theta, \theta, \theta), (\theta, 1 - \varepsilon, \theta, \theta), (\theta, \theta, 1 - \varepsilon, \theta), (\theta, \theta, \theta, 1 - \varepsilon)]$, whereas the fine weighting scheme was $\mathbf{W} = [(1 - \varepsilon, \theta, \theta, \theta), (\theta, 1 - \varepsilon, \theta, \theta,), (\theta, \theta, 1 - \varepsilon, \theta), (\theta, \theta, \theta, 1 - \varepsilon)$, (0.7 0.1 0.1 0.1), (0.4 0.2 0.2 0.2), (0.1 0.7 0.1 0.1), (0.2 0.4 0.2 0.2), (0.1 0.1 0.7 0.1), (0.2 0.2 0.4 0.2), (0.1 0.1 0.1 0.7), (0.2 0.2 0.2 0.4), (0.25 0.25 0.25 0.25)]. For each of the three houses, both weighting schemes yielded the same nondominated set of blockmodels. The computation times for the multiobjective tabu search algorithm when using the coarse weighting scheme were 0.18, 0.18, and 0.17 seconds for Houses A, B, and C, respectively. For the fine weighting scheme, the computation times were 0.14, 0.15, and 0.16 seconds for Houses A, B, and C, respectively.

In conjunction with the results for Sampson's (1968) monastery data in the previous section, these results suggest that, for small network matrices, the multiobjective tabu search algorithm is not overly sensitive to the choice of weights. For each house, the objective function values and cluster assignments for each of the blockmodels in the nondominated set are shown in Table 3. The table also highlights (in bold) the blockmodels that would be obtained via aggregation of the four network relations into a single network matrix.

The number of blockmodels in the nondominated sets for House A, House B, and House C are 5, 16, and 6, respectively. The ideal point for House A is ($f_1^* = 17$, $f_2^* = 15$, $f_3^* = 13$, $f_4^* = 15$). Based on the Euclidean distances of the objective functions for each blockmodel to this point, blockmodel A-3 is the best choice with a distance of 2.45. This blockmodel is also the one obtained by fitting a blockmodel to an aggregation of the four network relations into a single network matrix (Doreian, 2008, p. 255). Similar results were obtained for House C, which has an ideal point of ($f_1^* = 13$, $f_2^* = 14$, $f_3^* = 18$, $f_4^* = 15$). Blockmodel C-3, which is also what was obtained by aggregating the four network relations (Doreian, 2008), has the minimum distance (3.74) from this ideal point. Thus, in the cases of Houses A and C, there are relatively few nondominated blockmodels, which is evident of a concordant structure among the four relations. Under such conditions, aggregation across multiple relations appears to offer a perfectly viable strategy.

By contrast, there was greater discordance among the network relations for House B, which resulted in a larger nondominated set. The principal source of discordance for House B is the roommate relation. Blockmodel B-14 yields the minimum objective function value of $f_3^* = 17$ for the roommate relation but very poor objective function values for the other three relations. Other blockmodels, such as B-2, yield exceptional objective function values for the dating, friendship, and weekend relations but a poor objective function value for the roommate relation.

The ideal point for House B is ($f_1^* = 18$, $f_2^* = 18$, $f_3^* = 17$, $f_4^* = 19$). Based on the Euclidean distances of the objective functions for each blockmodel to this point, blockmodel B-3 is the best choice with a distance of 6.63. This blockmodel is one of two equally well-fitting partitions obtained when applying a generalized blockmodeling algorithm to an aggregation of the four network relations into a single network matrix (Doreian, 2008). The other equally well-fitting partition from the aggregation approach is blockmodel B-2, which has an appreciably greater distance of 9.06 from the ideal point. In fact, blockmodels B-4 and B-5 are closer to the ideal point than B-2. Thus, unlike Houses A and C where fitting a blockmodel to an aggregation of the relations resulted in arguably the best-fitting blockmodel in the nondominated set, the satisfactoriness of the aggregation approach is more ambiguous for House B. The results gleaned from the multiobjective tabu search heuristic indicate that one of the two equally well-fitting partitions (B-3) for the aggregated relations is an exceptionally good solution from the nondominated set, whereas the other (B-2) is not. Again, having the efficient frontier permits a more informed choice from a set of blockmodels. This suggests that the usual measures regarding the consistency of multiple relations are limited when one relation differs enough from the other indicators of the underlying signed relationship.

5. Example 3. Turning Point Project

5.1. *The Network Matrices*

For our third example, we demonstrate multiobjective blockmodeling using network data related to the Turning Point Project (TPP).[10] There were 108 organizations that collectively

[10] These data were collected and assembled by the second author.

TABLE 3.
Results for Example 2—Lemann and Solomon's (1952) dormitory data. The objective function values for the approximation of the Pareto efficient (nondominated) set of blockmodels. The results highlighted in bold are what would be obtained by fitting a blockmodel to an aggregation of the four network relations.

Block-model #	$f_1(B)$	$f_2(B)$	$f_3(B)$	$f_4(B)$	Distance from ideal	Cluster assignments
A-1	17	19	17	20	7.55	{1, 2, 4, 5, 6, 9, 10, 13, 16, 17} {7, 8, 14, 21} {3, 11, 18, 19} {12, 15, 20}
A-2	18	18	13	18	4.36	{1, 2, 4, 5, 6, 9, 10, 11, 13, 16, 17, 19} {7, 8, 15, 21} {3, 18} {12, 14, 20}
A-3	**19**	**16**	**13**	**16**	**2.45**	**{1, 2, 4, 5, 6, 9, 10, 11, 13, 16, 17} {7, 8, 15, 21} {3, 18, 19} {12, 14, 20}**
A-4	21	17	18	15	6.71	{1, 2, 5, 6, 9, 10, 11, 13, 16, 17} {7, 8, 15, 21} {3, 18, 19} {4, 12, 14, 20}
A-5	26	15	15	20	10.49	{1, 2, 4, 5, 6, 10, 11, 13, 16, 17} {7, 8, 15, 21} {3, 18, 19} {9, 12, 14, 20}
B-1	18	21	29	21	12.53	{1, 2, 4, 5, 8, 11, 12, 16} {6, 10} {3, 15} {7, 9, 13, 14, 17}
B-2	**19**	**18**	**26**	**19**	**9.06**	**{1, 2, 4, 5, 8, 9, 11, 12, 13, 16} {6, 10} {3, 15} {7, 14, 17}**
B-3	**20**	**20**	**23**	**19**	**6.63**	**{1, 2, 4, 5, 6, 8, 11, 12, 13, 16} {9, 10} {3, 15} {7, 14, 17}**
B-4	20	24	22	21	8.31	{1, 2, 4, 5, 8, 11, 12, 13, 16} {6} {3, 15} {7, 9, 10, 14, 17}
B-5	21	19	25	21	8.83	{1, 2, 4, 5, 8, 9, 11, 12, 16} {6, 10} {3, 13, 15} {7, 14, 17}
B-6	23	25	21	24	10.72	{1, 2, 4, 5, 6, 8, 12, 13, 16} {9, 10} {3, 15} {7, 11, 14, 17}
B-7	23	27	18	25	11.96	{1, 2, 4, 5, 8, 12, 13, 16} {6, 11} {3, 15} {7, 9, 10, 14, 17}
B-8	23	29	19	24	13.23	{1, 2, 4, 5, 6, 8, 12, 13, 16} {11} {3, 15} {7, 9, 10, 14, 17}
B-9	24	23	21	22	9.27	{1, 2, 4, 5, 8, 12, 13, 16} {6, 9, 10, 11} {3, 15} {7, 14, 17}
B-10	24	26	20	22	10.86	{1, 2, 4, 5, 6, 8, 12, 13, 16} {9, 10, 11} {3, 15} {7, 14, 17}
B-11	26	24	21	19	10.77	{1, 2, 3, 4, 5, 8, 12, 13} {6, 10, 11, 16} {15} {7, 9, 14, 17}
B-12	27	20	22	20	10.54	{1, 2, 3, 4, 5, 8, 9, 12, 13} {6, 10, 11, 16} {15} {7, 14, 17}
B-13	27	21	20	19	9.95	{1, 2, 3, 4, 5, 8, 12, 13} {6, 9, 10, 11, 16} {15} {7, 14, 17}
B-14	27	25	17	22	11.79	{1, 2, 3, 4, 5, 8, 12, 13} {6, 11, 16} {15} {7, 9, 10, 14, 17}
B-15	28	26	19	21	13.11	{1, 2, 3, 4, 5, 8, 12, 13} {11, 16} {6, 15} {7, 9, 10, 14, 17}
C-1	13	19	27	21	11.92	{18} {4, 6, 9, 10, 13, 19} {2, 3, 7, 11, 12, 16, 17} {1, 5, 8, 14, 15, 20}
C-2	15	17	24	18	7.62	{10} {4, 6, 9, 13, 19} {2, 3, 7, 11, 12, 16, 17} {1, 5, 8, 14, 15, 18, 20}
C-3	**16**	**14**	**19**	**17**	**3.74**	**{1, 20} {4, 6, 9, 10, 13, 19} {2, 3, 7, 11, 12, 16, 17} {5, 8, 14, 15, 18}**
C-4	17	18	22	15	6.93	{20} {4, 6, 9, 10, 13, 19} {2, 3, 7, 11, 12, 16, 17} {1, 5, 8, 14, 15, 18}
C-5	19	15	20	15	6.40	{20} {4, 6, 9, 10, 13, 19} {2, 3, 7, 11, 12, 16, 17} {1, 5, 8, 14, 15, 18}
C-6	20	17	18	15	7.62	{10} {4, 6, 9, 13, 19} {1, 2, 3, 7, 11, 12, 16, 17, 20} {5, 8, 14, 15, 18}

signed some radical advertisements regarding the environment in the New York Times (NYT) during 1999–2000 where different subsets of organizations signed different advertisements. It is reasonable to think that organizations signing advertisements together would show some consistency with those organizations sharing board members. When the board data were collected from their websites, it was clear that only 64 of them were involved in sharing board members. The following analysis is restricted to these 64 organizations. There are two network matrices associated with this application: (1) a shared-board network matrix and (2) an advertisement signing matrix. The first of these ($\mathbf{A}_1$) is a board interlock matrix of ties among $n = 64$ organizations participating in the TPP and involved in sharing board members. A matrix element $a_{ij1} = 1$ if organization i shares board member relationships with organization j and $a_{ij1} = 0$ otherwise. The second matrix ($\mathbf{A}_2$) is a 64×25 two-mode matrix that reflects ties between organizations and full one-page advertisements in the NYT that they signed. More specifically, elements of this two-mode matrix assume values of $a_{ij2} = 1$ if organization i signed advertisement j and $a_{ij2} = 0$ otherwise. The 25 advertisements were in five categories defined by the organizers of the TPP. These categories were: Ecological catastrophe (EC), Economic globalization (EG), Genetic Engineering (EG), Industrial agriculture (IA), and Techno-mania (TM). We have taken this classification as given by the TPP for the following analysis. These categories provide insight into the natural or logical partitioning of the advertisements into underlying themes.

5.2. *Short- vs. Long-Term Common Interests*

This application of multiobjective blockmodeling is grounded by some assumptions regarding common interests among organizations in both the long and short terms. For example, if joint board memberships signal common interests, then these are long-term interests that are aligned given that board memberships are most often durable. The signing of ads is more likely to mobilize short-term interests. Nevertheless, as noted above, it is reasonable to anticipate some correspondence between the joint board memberships and the signing of ads, yet the extent is unclear. Multiobjective blockmodeling is a useful tool for examining the extent to which there is some degree of correspondence. We seek to find a partition of the 64 organizations that produces a well-fitting blockmodel for both the board interlock and ad signing network matrices. The objective function for the board interlock network matrix is

$$f_1(B) = \sum_{k=1}^{K} \sum_{(i \neq j) \in S_k} (1 - a_{ij1}) + \sum_{k=1}^{K-1} \sum_{l=k+1}^{K} \sum_{i \in S_k} \sum_{j \in S_l} (a_{ij1} + a_{ji1}). \tag{8}$$

Thus, the objective function for the board interlock network is to minimize the sum of the absence of joint board membership ties within clusters, plus the sum of interlocked board ties that occur between clusters.

The objective function for the signed ad network is to minimize the total number of inconsistencies with an ideal block structure of null and complete blocks. More specifically, the objective function is the sum of 1s that occur in blocks that have mostly 0s, plus the sum of 0s that occur in blocks that have mostly 1s. This objective function is concordant with those used in recent applications of two-mode blockmodeling (Brusco & Steinley, 2011; Brusco, Doreian, Steinley, & Mrvar, in press; Doreian, Batagelj, & Ferligoj, 2004). As noted previously, we assume that the ads are partitioned into $L = 5$ five natural categories ($C_1, C_2, \ldots, C_5$) defined by an underlying theme.[11] The objective function for the signed ad network matrix is

$$f_2(B) = \sum_{k=1}^{K} \sum_{l=1}^{L} \min\{\lambda_{kl}, \rho_{kl}\}, \tag{9}$$

[11] We conducted extensive single-objective two-mode blockmodeling analyses that verified that this natural partition of the ads was, in most cases, also the most appropriate partition of the ads from an analytical standpoint.

where

$$\lambda_{kl} = \sum_{i \in S_k} \sum_{j \in C_l} a_{ij2} \quad \forall 1 \le k \le K \text{ and } 1 \le l \le L, \tag{10}$$

and

$$\rho_{kl} = \sum_{i \in S_k} \sum_{j \in C_l} (1 - a_{ij2}) \quad \forall 1 \le k \le K \text{ and } 1 \le l \le L. \tag{11}$$

By defining w_1 and w_2 as objective function weights for the board interlock and signed ads network matrices, respectively, the single-objective, weighted-sum optimization problem is

$$\text{Minimize:} \qquad F(\mathbf{w}, B) = \sum_{g=1}^{2} w_g f_g(B), \tag{12}$$

$$\text{subject to} \qquad B \in \boldsymbol{\mathcal{B}}. \tag{13}$$

5.3. Results

We began by applying 100 restarts of the bond-energy algorithm (McCormick, Schweitzer, & White, 1972) to obtain a permuted version of $\mathbf{A}_2$ that would facilitate a reasonable choice of K. Visual inspection of the permuted matrix suggested the appropriateness of an 8-cluster model.[12] It was not possible to apply an exact solution method to obtain optimal solutions to the single-objective problems, and, therefore, we used a relocation heuristic algorithm to obtain the single-objective blockmodels used to initially populate the nondominated set. Next, the multiobjective tabu search heuristic was applied to the network matrix for $K = 8$ using the same two weighting schemes described in Section 3.3. The coarse weighting scheme, $\mathbf{W} = [(0.999, 0.001), (0.001, 0.999)]$, yielded a nondominated set consisting of 73 blockmodels, whereas the fine weighting scheme, $\mathbf{W} = [(0.999, 0.001), (0.9, 0.1), (0.8, 0.2), (0.7, 0.3), \ldots (0.3, 0.7), (0.2, 0.8), (0.1, 0.9), (0.001, 0.999)]$, produced 77 blockmodels. The total computation times for the multiobjective tabu search heuristic were 100.70 and 100.53 seconds for the coarse and fine weighting schemes, respectively.

We conducted a "cross-checking" of the results by comparing the approximated nondominated sets from the two weighting schemes to one another and discarding any blockmodel in one set that is dominated by one or more blockmodels in the other. When comparing the blockmodels obtained via the two-weighing schemes, it was found that 52 of the 73 blockmodels obtained for the coarse weighting scheme were actually nondominated. Contrastingly, all 77 of the blockmodels obtained with finer granularity in the weights were nondominated. The conclusion from this finding is that finer granularity can afford better estimation of the nondominated set for larger network matrices, particularly when there is discordance among the objective criteria. One approach that can be applied is to begin with a coarse weighing scheme and generate the nondominated set. If a satisfactory blockmodel is not identified, then the analyst can interactively use the method by restarting the algorithm with an expanded (finer) set of weights. For example, restarting the algorithm using the fine weighting scheme, but with the coarse weighting scheme blockmodels to initially populate the nondominated set, resulted in an increase in the number of blockmodels that were actually nondominated from 52 to 73.

The Pareto efficient frontier (using the finer granularity weighting scheme) for the TPP network is provided in Figure 2. Because of the disparity in the raw objective function values, Figure 2 displays the nondominated set in terms of percentage deviation above the ideal point.

[12] The selection of K is an important decision, and we acknowledge that there are other approaches that might facilitate its selection. However, methods for choosing the number of clusters are not the focal point of this paper, and we deemed $K = 8$ clusters to be a reasonable choice that met the goals of our illustration.

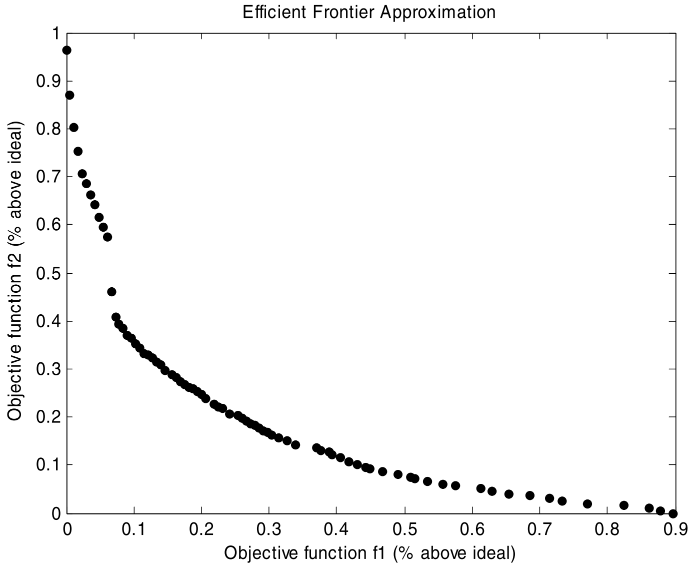

FIGURE 2.
The approximation of the Pareto efficient (nondominated) set for the Turning Point Project networks.

The ideal point for the network matrix is ($f_1^* = 392$, $f_2^* = 165$). The minimization of one criterion to the exclusion of the other produces an extremely poor objective function value for the other criterion. For example, the partition of organizations yielding $f_1 = 392$ produces a value of $f_2 = 321$ (95 % greater than the ideal value). Similarly, the partition of organizations yielding $f_2 = 165$ produces a value of $f_1 = 838$ (114 % greater than the ideal value). The big differences in the single-objective solutions principally stem from the f_1 criterion exhibiting a propensity for similarly sized clusters, whereas the f_2 criterion is more amenable to differences in the cluster sizes. At face value, the results suggest considerable antagonism between the two objective criteria. This is important because it might point to a real difference between long-term interests (in terms of shared board memberships) and short-term interests (in the signing of specific ads). However, the sharply bending curve in Figure 2 reveals that very small sacrifices in f_1 will produce substantial improvements in f_2 and vice versa. This insight suggests that, even with antagonism between a pair of objective criteria, multiobjective blockmodeling is sufficiently flexible to handle this antagonism in a useful fashion.

Based on the criterion of Euclidean distance of the blockmodel from the ideal point (on a percentage basis), we selected a blockmodel with objective function values of $f_1 = 486$ and $f_2 = 201$, which are 24 % and 22 % greater than their ideal values, respectively. A visual representation of the blockmodel for the signed ads network is displayed in Figure 3. Figure 4 contains the number of inconsistencies that occur within and between clusters for the board interlock matrix. Along the main diagonal of Figure 4 is the sum of the total number of absences of board ties among organizations in that cluster. The entries off the main diagonal contain the number of ties that occur between organizations in two different clusters. Each cell also expresses (in parentheses) the number of inconsistencies as a percentage of the total possible ties that could have occurred. The percentages along the main diagonal seem large at first glance, averaging 67.2 %. However, these figures should be considered in light of the fact that the sparseness of the board interlock matrix results in a minimum possible percentage average of 55 %.

	EC ads				GE ads					IA ads						EG ads					TM ads				
"eii"	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1
"foe"	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
"gpus"	1	1	1	0	0	1	1	0	1	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0
"icta"	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1
"ran"	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0	1	1	1	1	1	1	0	1	1	1
"fde"	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
"ala"	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
"cbd"	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
"dw"	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
"esc"	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
"nfpa"	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
"nfn"	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
"rnw"	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
"wp"	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
"cfs"	0	0	0	0	1	1	1	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
"crg"	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
"faw"	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
"ff"	0	0	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
"iatp"	0	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	0	0	0	0	0
"isec"	0	0	0	0	0	1	1	1	0	1	0	1	1	1	0	1	0	1	0	0	1	0	1	0	0
"oca"	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
"pan"	0	0	0	0	0	1	1	0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0
"cua"	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
"esi"	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
"gi"	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
"li"	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0
"ofrf"	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
"ri"	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
"scp"	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
"50fyi"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0
"coc"	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0
"ge"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
"ifg"	0	1	0	0	1	0	1	0	1	1	1	1	1	0	0	1	1	1	1	1	0	1	0	1	0
"pcdf"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
"pi"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
"rfste"	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0
"uswa"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
"bchi"	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	1	1
"cmd"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
"em"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0
"jes"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0
"loka"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1
"ni"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
"svtc"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1
"wtw"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
"yes"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	1	1	0
"cccp"	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
"carp"	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0
"fa"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
"imc"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
"ilsr"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
"icrt"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
"iffa"	0	0	0	0	1	1	1	0	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0
"pmc"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
"rcfc"	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
"amf"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
"afc"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
"cem"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
"ei "	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
"grac"	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
"nfc"	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0
"onda"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
"tikkm"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	1
"tvto"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0

FIGURE 3.
Blockmodel of 2-mode Turning Point Project network.

	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$
$k=1$	20 (55.56)	2 (4.17)	2 (4.17)	1 (2.38)	5 (10.42)	7 (12.96)	0 (0.00)	3 (5.56)
$k=2$	–	44 (68.75)	1 (1.56)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
$k=3$	–	–	48 (75.00)	6 (10.71)	10 (15.63)	5 (6.94)	1 (1.39)	6 (8.33)
$k=4$	–	–	–	34 (69.39)	0 (0.00)	2 (3.17)	0 (0.00)	2 (3.17)
$k=5$	–	–	–	–	36 (56.25)	3 (4.17)	0 (0.00)	2 (2.78)
$k=6$	–	–	–	–	–	58 (71.60)	2 (2.47)	6 (7.41)
$k=7$	–	–	–	–	–	–	58 (71.60)	0 (0.00)
$k=8$	–	–	–	–	–	–	–	56 (69.14)

FIGURE 4.

Inconsistencies for the board interlock matrix. Each cell contains the number of inconsistencies observed for each pair of clusters. Along the main diagonal, the entries are the total absences of board ties among organizations within the same cluster. The above-diagonal elements are the total board ties between organizations in different clusters (the lower triangle of the matrix is the mirror image of the upper triangle). The percentage of inconsistencies relative to the total number of elements is shown in parentheses.

The first cluster consists of six organizations[13] ("eii", "foe", "gpus", "icta", "ran", "fde") that share a total of four joint board membership ties among them. Most notably, "fde" and "eii" share board memberships with one another and are each linked to two other organizations in the cluster. Figure 3 reveals that these six organizations are linked to the signing of four of the five groups of advertisements (EC, GE, EG, and TM). The second cluster consists of eight organizations ("ala", "cbd", "dw", "esc", "nfpa", "nfn", "mw", "wp") that tend to sign only ads in the EC group.[14] The organizations in this cluster share a total of six joint board memberships among them, and "mw" contributes three of these ties.

Cluster three is the only other cluster (besides cluster one) that heavily signs ads in more than one group. The third cluster consists of eight organizations ("cfs", "crg", "faw", "ff", "iatp", "isec", "oca", "pan") that are heavy signers of ads in both the GE and IA groups. The organizations in this cluster share a total of four joint board memberships among them, mainly stemming from "cfs". Clusters four, five, and six are heavy signers of ads in groups IA, EG, and TM, respectively. Cluster five is especially noteworthy among these, as it is a cluster of eight organizations that share a total of 10 joint board memberships. Accordingly, the organizations in this cluster seem especially concordant in terms of sharing both long-term (joint board membership) and short-term (ad signing) ties. The seventh and eighth clusters each share a reasonable degree of joint board membership ties; however, neither of these two clusters consistently endorsed ads in a particular category.

The analysis of the TPP network data reveals that, again, the multiobjective tabu search heuristic provides a solution that minimizes the distance from the ideal point when the objectives

[13] The labeling of these organizations uses the first initials of their full names. However, their identities are not important for the analysis presented here.

[14] We note that 'eii' and 'fde' also sign ads in the IA area while the other organizations in this cluster sign none of these ads.

are considered simultaneously and breaks ties between seemingly comparable blockmodels. Additionally, the analysis emphasizes the importance of granularity in approaching large network problems. Finally, multiobjective blockmodeling offers the analyst a way to reconcile and solve problems that include highly antagonistic networks.

6. Simulation Results and Recommendations for Practice

6.1. Simulation Study 1

The evaluation of multiobjective programming methods is nontrivial (Erhgott & Gandibleaux, 2000). The consistency and interpretability of the results obtained by the multiobjective tabu search heuristic for blockmodeling in the examples provided in Sections 3, 4, and 5 offer some positive evidence of its effectiveness. To further substantiate the merits of the multiobjective tabu search approach for blockmodeling, we conducted a simulation study to ensure that the method performs as intended for problems where at least a portion of the nondominated set is known with certainty.

The context of the simulation study was biobjective blockmodeling of a signed network matrix based on structural balance theory, the same problem considered in Section 3. We generated 20 signed network matrices of size 20 × 20. All test problems were generated under the assumption of four clusters, each containing five objects. For the first 10 problems, the probability of a positive tie between two objects in the same cluster was 60 %, the probability of a negative tie was 10 %, and the probability of no tie was 30 %. Similarly, the probability of a negative tie between two objects in different clusters was 60 %, the probability of a positive tie was 10 %, and the probability of no tie was 30 %. For Problems 11 through 20, the probabilities of the correct/incorrect signs within and between clusters were changed from 60 %/10 % to 50 %/20 %, and, therefore, these latter 10 problems were less well structured and would presumably have a larger nondominated set.

Multiobjective blockmodeling solutions for each of the 20 test problems were obtained under four different versions for the heuristic: (a) a coarse weighting scheme of $\mathbf{W} = [(0.999, 0.001), (0.001, 0.999)]$ and using a random blockmodel to initially populate the nondominated set, (b) a fine weighting scheme of $\mathbf{W} = [(0.999, 0.001), (0.9, 0.1), (0.8, 0.2), \ldots, (0.2, 0.8), (0.1, 0.9), (0.001, 0.999)]$ and using a random blockmodel to initially populate the nondominated set, (c) the coarse weighting scheme using the single-objective optimal solutions to initially populate the nondominated set, and (d) the fine weighting scheme using the single-objective optimal solutions to initially populate the nondominated set. For each test problem and each of the four versions, data were collected with respect to both the approximated nondominated set and computation time. For each test problem, the nondominated sets obtained under the four versions were compared to one another, and any blockmodels that were dominated by a blockmodel obtained under one or more of the other versions were discarded. A summary of the results is provided in Table 4.

The results in Table 4 reveal that the computation times for multiobjective tabu search algorithm were very modest, regardless of the test problem characteristics, weighting scheme, and initial population of the nondominated set. It is particularly interesting to observe that, despite the fact that the sizes of the nondominated sets for Problems 11 through 20 were much larger than those for Problems 1 through 10, the computation times were comparable across these two sets of problems. Most of the computation times were less than one second. By contrast, much greater computation time variability is observed for the branch-and-bound algorithm used to obtain the single-objective optimal blockmodels for populating the initial nondominated set for versions (c)

TABLE 4.

Results for 20 test problems in simulation Study 1. The "B&B" columns contains the computation time for the branch-and-bound algorithm to obtain the single-objective optimal blockmodels used to initially populate the nondominated set. The four versions of the multiobjective blockmodeling algorithm are: (a) weighting scheme of $[1, 0; 0, 1]$ and using a random blockmodel to initially populate the nondominated set, (b) weighting scheme of $[1, 0; 0.9, 0.1; 0.8, 0.2; \ldots 0, 1]$ and using a random blockmodel to initially populate the nondominated set, (c) weighting scheme of $[1, 0; 0, 1]$ and using the single-objective optimal solutions to initially populate the nondominated set, and (d) weighting scheme of $[1, 0; 0.9, 0.1; 0.8, 0.2; \ldots 0, 1]$ and using the single-objective optimal solutions to initially populate the nondominated set. Values in bold type in the last four columns indicate that all of the supported Pareto efficient blockmodels were found.

Test problem	Computation times					Number of blockmodels in nondominated set				Number of "supported" Pareto efficient blockmodels			
	B&B	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
1	0.79	0.34	0.76	0.56	0.75	17	17	17	15	**4**	**4**	**4**	**4**
2	0.21	1.11	0.48	1.00	0.24	5	13	13	10	*1*	**2**	**2**	**2**
3	0.07	0.37	0.40	0.42	0.53	10	9	12	11	*1*	*1*	**2**	**2**
4	0.78	0.48	0.21	0.25	0.45	5	4	10	11	*1*	*1*	**3**	**3**
5	0.44	0.42	0.17	0.30	0.15	9	3	6	6	**3**	*2*	**3**	**3**
6	0.30	0.35	0.36	0.30	0.36	9	3	7	6	**3**	*2*	**3**	**3**
7	0.26	0.28	0.16	0.18	0.26	4	4	8	10	*2*	*2*	**3**	**3**
8	0.47	0.54	0.42	0.99	0.31	12	9	13	14	*3*	*3*	**4**	**4**
9	0.37	0.39	0.28	0.60	0.31	14	15	14	8	**3**	2	**3**	**3**
10	5.83	0.45	0.31	0.25	0.47	17	13	16	15	**4**	3	**4**	**4**
11	26.10	1.17	0.61	0.27	0.53	35	35	35	35	**7**	**7**	**7**	**7**
12	5.03	1.50	1.12	0.99	0.93	22	26	24	24	**5**	**5**	**5**	**5**
13	3.73	1.40	0.52	0.88	0.50	33	33	33	33	**5**	**5**	**5**	**5**
14	22.87	0.61	0.61	0.70	0.45	25	26	25	25	**4**	**4**	**4**	**4**
15	3.35	1.22	0.92	1.53	0.55	29	30	30	31	**6**	**6**	**6**	**6**
16	10.25	0.65	0.42	0.67	0.42	29	29	29	29	**5**	**5**	**5**	**5**
17	15.92	1.22	0.69	0.86	0.50	37	38	38	38	**4**	**4**	**4**	**4**
18	16.31	0.91	0.40	0.91	0.40	33	33	33	33	**5**	**5**	**5**	**5**
19	58.54	2.23	0.83	1.35	0.96	33	33	33	33	**5**	*4*	**5**	**5**
20	19.67	0.78	0.55	0.75	0.53	29	30	30	30	**5**	**5**	**5**	**5**

and (d). The branch-and-bound algorithm is sensitive to the data characteristics. The computation times for Problems 11 through 20 are typically greater because they are less well structured (i.e., have more inconsistencies) than Problems 1 through 10.

Because the test problems are limited to $n = 20$ objects, we were able to use the branch-and-bound algorithm under different weighting schemes (i.e., $[(0.999, 0.001), (0.9, 0.1), (0.8, 0.2), \ldots, (0.2, 0.8), (0.1, 0.9), (0.001, 0.999)]$) to obtain "supported" Pareto efficient solutions. This enabled us to obtain an objective measure of how well each of the four versions performed with respect to the recovery of blockmodels that are guaranteed to be supported Pareto efficient solutions in the dominated set. Heuristic versions (d) and (c) performed exceptionally well, finding all of the supported Pareto efficient solutions for all 20 test problems, respectively. Versions (a) and (b) failed to obtain all of the Pareto efficient solutions for a total of 5 and 7 problems, respectively. With one exception (version (b) on Problem 19), all four versions obtained all Pareto efficient solutions for Problems 11 through 20 and were always comparable in terms of the size of the approximated nondominated set for these problems. The different versions of the heuristic exhibited more variability in their nondominated sets for Problems 1 through 10. In light of the failure of versions (a) and (b) to identify the complete set of Pareto efficient solutions for several of these problems, we conclude the initial population of the nondominated set is a vital component of our methodology. There was no pronounced difference in performance between version

TABLE 5.

Results for 20 test problems in simulation Study 2. The “RH” column contains the computation time for 1000 restarts of the relocation heuristic used to obtain the single-objective optimal blockmodels used to initially populate the nondominated set. The two heuristic versions correspond to coarse weights: $[(0.999, 0.001), (0.001, 0.999)]$ and fine weights $[(0.999, 0.001), (0.9, 0.1), (0.8, 0.2), \dots (0.2, 0.8), (0.1, 0.9), (0.001, 0.999)]$. The term “cross-checking” refers to comparison of the approximated nondominated sets from the two weighting schemes to one another and the discarding of any blockmodel in one set that is dominated by one or more blockmodels in the other.

Problem	Computation times			Number of blockmodels in the approximated Pareto efficient set prior to “cross-checking”		Number of blockmodels in the approximated Pareto efficient set after “cross-checking”	
	RH	Coarse	Fine	Coarse	Fine	Coarse	Fine
1	6.05	215.36	154.12	268	272	68	231
2	5.82	178.24	151.22	289	313	130	300
3	5.79	215.63	99.57	277	286	254	284
4	5.83	174.39	88.49	259	250	224	157
5	5.97	233.74	124.96	263	266	210	258
6	5.95	234.79	156.92	277	280	213	267
7	6.00	249.47	99.25	273	279	70	272
8	5.90	272.67	80.65	262	238	120	183
9	5.76	124.68	71.10	249	239	193	177
10	5.87	130.02	80.74	204	217	191	183
11	9.25	463.99	260.88	562	571	354	489
12	8.98	323.67	176.15	574	570	532	561
13	9.82	257.41	264.99	621	617	524	611
14	9.42	561.64	307.19	701	730	598	689
15	8.78	322.41	205.33	595	601	526	595
16	8.89	471.20	272.00	633	659	534	622
17	9.67	414.46	215.22	564	568	488	550
18	9.18	593.23	238.29	680	673	385	620
19	9.10	678.23	229.14	634	640	616	606
20	9.11	615.94	280.88	624	622	555	562

(c) and (d), and, accordingly, the coarse weighting scheme was sufficient for these smaller problems.

6.2. *Simulation Study 2*

To determine whether the required granularity of the weighting scheme might be a function of problem size, we repeated simulation Study 1 after increasing the number of objects from $n = 20$ to $n = 100$. All test problems were generated under the assumption of four clusters, each containing 25 objects. The same probabilities used in Study 1 were used for data generation to produce 20 test problems. The nondominated set was initially populated using 1000 restarts of a relocation heuristic for the single-objective problems. The multiobjective tabu search heuristic was then applied using two different weighting schemes: (1) coarse, $\mathbf{W} = [(0.999, 0.001), (0.001, 0.999)]$, and (2) fine, $\mathbf{W} = [(0.999, 0.001), (0.9, 0.1), (0.8, 0.2), \dots, (0.2, 0.8), (0.1, 0.9), (0.001, 0.999)]$. The two weighting schemes were compared in terms of computation time, as well as the size of the nondominated sets after cross-checking the results of the two weighting schemes and discarding dominated blockmodels. The results are reported in Table 5.

There are three important aspects with respect to the computation time results in Table 5. First, the computation times for the multiobjective tabu search algorithm are drastically larger than those in Table 4, which indicates that it is sensitive to the size of the network. Second, the computation times for Problems 11 through 20 are roughly two to three times greater than those for Problems 1 through 10, which suggests that the form of structure in the network affects computational time and network size. Third, the computation times for the fine weighting scheme are, in most cases, appreciably less than those for the coarse weighting scheme. At first glance, this seems counterintuitive. However, it is attributable to the fact that the coarse weighting scheme typically fleshes out the nondominated set more slowly than the fine weighting scheme. Once several cycles at each set of weights in the fine weighting scheme have occurred, the frontier is generally well approximated. Contrastingly, with the coarse scheme, the algorithm must make many cycles, gradually adding nondominated blockmodels to the frontier.

The results in Table 5 also reveal that a finer weighting scheme can often afford significant improvements in the approximated nondominated set for larger problems. The coarse and fine weighting schemes initially establish comparably sized nondominated sets. However, after cross-checking between these two sets, the nondominated set for the fine weighting scheme is larger for 16 of the problems and very close in number to the coarse weighting scheme on three of the remaining four problems (i.e., Problems 9, 10, and 19). By contrast, cross-checking often substantially reduces the number of blockmodels in the nondominated set for the coarse weighting scheme, sometimes by more than 50 %. Problems 1, 2, 7, 11, and 18 are among those where the nondominated set of the coarse weighting scheme is severely pruned by the cross-checking process. Consider, for example, Problem 2, where the initial sizes of the nondominated set are comparable at 289 and 313 for the coarse and fine weighting schemes, respectively. After cross-checking, only 13 of the blockmodels in the fine set are dominated by blockmodels in the coarse set, thus reducing the number for the fine weighting scheme from 313 to 300. By comparison, 159 blockmodels in the coarse set are dominated by blockmodels from the fine set, thus reducing the number for the coarse weighting scheme from 289 to 130.

6.3. *Recommendations for Practice*

The results of the first two numerical illustrations and simulation Study 1 suggest that the multiobjective tabu search algorithm for blockmodeling performs comparably for both fine and coarse weighting schemes. The finer weighting scheme only offered a significant advantage in the cases of the Turning Point Project example and simulation Study 2, which are larger networks with greater discordance between the objective functions. The results of both simulation studies and the Turning Point Project example indicate that there are distinct improvements in the nondominated set that can be realized by initially populating the nondominated set with (near-)optimal blockmodels for each objective function independently. Based on these results, we recommend the following steps for an analyst using the applied multiobjective blockmodeling procedure.

Step 1. When possible, obtain an optimal blockmodel for each objective function independently and use these to initially populate the nondominated set. If this is not practical, then use a heuristic procedure to obtain near-optimal blockmodels for the initial population.

Step 2. Run the multiobjective tabu search algorithm using the coarse weighting scheme $\mathbf{W} = [(1-\varepsilon, \theta, \dots, \theta), (\theta, 1-\varepsilon, \theta, \dots, \theta), \dots, (\theta, \dots, \theta, 1-\varepsilon)]$, where ε is a small positive constant ($0 < \varepsilon \leq 0.001$), and $\theta = \varepsilon/(G-1)$. In other words, there are G sets of weights, each associated with a weight of $(1-\varepsilon)$ for one criterion and an equal spread of weights of $\theta = \varepsilon/(G-1)$ among the other $G-1$ criteria. In our applications, using $\varepsilon = 0.001$ was always sufficient to ensure that no improvement in the $G-1$ criteria weighted at θ would be worth a sacrifice of the criterion weighted $(1-\varepsilon)$.

Step 3. The analyst's preferences for tradeoffs among the competing criteria are applied to the approximate nondominated set obtained in Step 2 to select a blockmodel. To facilitate this process, the analyst could be presented with Euclidean distances from the ideal point, perhaps after some normalization of the objective functions (see Brusco & Stahl, 2001).

Step 4. If the analyst identifies a suitable blockmodel (or set of blockmodels) in Step 3, then the selection process is complete. Otherwise, the multiobjective tabu search algorithm can be reapplied using the approximated nondominated set (or, possibly, a select region of that set) from Step 2 as the initial population, as well as an expansion of the weighting scheme to a finer set of weights. The updated frontier is then reevaluated via a return to Step 3.

7. Conclusions

7.1. Summary

We presented a multiobjective programming model for blockmodeling of network matrices. A general multiobjective tabu search algorithm was offered as a plausible method for direct approximation of the Pareto efficient frontier. Among the benefits of our proposed model and method are: (1) the ability to obtain blockmodels that fit well with respect to multiple blockmodeling criteria, particularly when relationships among those criteria are antagonistic, (2) mitigation of the problem of multiple equally well-fitting partitions, (3) avoidance of the need to prespecify weights for different objective functions. The multiobjective programming model and method can be adapted for many different types of blockmodeling applications. In this paper, we have reported results for three applications. The first application (Monastery data) was to a well-known signed network corresponding to affect ties among 18 monks at a monastery (Sampson, 1968). Using the sum of negative affect ties within groups and the sum of positive affect ties between groups as the objective criteria, the multiobjective tabu search algorithm was applied to the network matrix. For this small network, the algorithm was insensitive to parameter choices, rapidly generating the Pareto efficient set of blockmodels and enabling the analyst to select blockmodels from the set without the need to prespecify a weighting scheme.

The second application (House data) focused on finding a partition of actors that yielded a good blockmodel fit for $G = 4$ distinct network relations. The relations pertain to double-dating, post-college friendship, potential roommate, and weekend outing ties among women in off-campus dormitories in an eastern college. Using multiobjective blockmodeling, it was possible to rapidly estimate the nondominated set of blockmodels and identify partitions that yield reasonable tradeoffs among the different relations.

The third application (TPP data) used multiobjective blockmodeling to find a partition of organizations that maximized joint board memberships within clusters, as well as shared signing of ads in the Turning Point Project. Thus, like the House data in the second application, the focus was on multiple network relations for the same objects. However, unlike the House data, the relationship between the two criteria in the TPP data was highly antagonistic. That is, the optimal partitions for the board interlock network yielded poor results for the signed ads network and vice versa. Nevertheless, multiobjective blockmodeling identified blockmodels that afford excellent tradeoffs between the two criteria.

7.2. Limitations

Among the inherent limitations of the multiobjective tabu search heuristic presented in this paper are: (1) the potential need to reset parameters, (2) the need to prespecify a weighting scheme, (3) the fact that the nondominated set is only approximated, and (4) the challenge of choosing solutions from the Pareto efficient set. The parameters selected seem fairly robust across different network sizes, objective functions, and objective function configurations. The weighting scheme selection is somewhat more important; however, we offer a structured approach in

Section 6.3. Like most heuristics for combinatorial optimization problems, the proposed multiobjective tabu search algorithm is not guaranteed to obtain globally optimal solutions, and, accordingly, we are careful to point out that, in general, the algorithm only approximates the Pareto efficient frontier. For our applications to Sampson's (1968) monastery data and Lemann and Solomon's (1952) dormitory data, we were able to use a previously published branch-and-bound algorithm (Brusco & Steinley, 2010) to verify the supported Pareto efficient solutions on the frontier, but this is typically not possible for larger networks. Finally, the selection of a blockmodel from the Pareto efficient set will almost certainly require some information that reflects the preferences of the analyst. We offer some simple measures of distance from the ideal point to help facilitate this task.

7.3. The Larger Framework of Multiobjective Clustering

Multiobjective blockmodeling can be perceived as a special case of multiobjective clustering (see Ferligoj & Batagelj, 1992, for an overview). Multiobjective clustering applications date back, at least, to the work of Delattre and Hansen (1980), who presented an exact algorithm for a biobjective partitioning problem with objectives of diameter and split. Brusco and Stahl (2005, Chapter 6) and Brusco and Steinley (2009b) discuss exact solution methods for biobjective clustering problems with within-cluster sum-of-squares objective functions for two distinct sets of clustering variables. These clustering applications are similar to those encountered in multiobjective market segmentation (Brusco, Cradit, & Stahl, 2002; Brusco, Cradit, & Taschchian, 2003; DeSarbo & Grisaffe, 1998), where simulated annealing approaches have been especially popular. More recently, genetic/evolutionary algorithms have proved successful in a variety of clustering contexts (Handl, Kell, & Knowles, 2007; Handl & Knowles, 2007; Liu, Ram, Lusch, & Brusco, 2010; Ripon, Tsang, Kwong, & Ip, 2006; Saha & Bandyopadhyay, 2010) and could potentially be adapted as a viable alternative for multiobjective blockmodeling. Accordingly, one possible extension to our paper is the design of alternative heuristic procedures for direct estimation of the Pareto efficient set and the subsequent comparison of their estimated frontiers to those obtained by our algorithm.

It is also important to observe that, just as other multiobjective programming approaches for clustering can be adapted for the blockmodeling problems studied herein, the multiobjective tabu search heuristic presented herein can be adapted for other clustering problems. For example, it might be interesting to compare an adaptation of the MOTS paradigm for multiobjective market segmentation to some of the simulated annealing, genetic algorithm, and local-search heuristic procedures used in previous studies (Brusco et al., 2002, 2003; DeSarbo & Grisaffe, 1998; Liu et al., 2010).

7.4. Extensions to Stochastic Blockmodeling

We have framed our coverage of multiobjective blockmodeling within the context of *deterministic* blockmodeling. Our multiobjective programming model formulation is clearly a discrete optimization problem, whereas *stochastic* blockmodeling procedures typically rely on continuous (or smooth) optimization procedures such as the Newton–Raphson method, the E–M algorithm (Dempster, Laird, & Rubin, 1977), Markov chain Monte Carlo estimation (Gilks, Richardson, & Spiegelhalter, 1996), or iteratively reweighted least squares (Wasserman & Pattison, 1996). Nevertheless, there is recent work in stochastic blockmodeling where it appears that researchers are now using combinatorial algorithms to optimize likelihood-based optimization criteria (Karrer & Newman, 2011; Steinley et al., 2011). In light of these developments, it seems possible that our proposed multiobjective programming based framework might hold some promise for stochastic blockmodeling as well. For example, suppose that two likelihood-based criteria were under consideration for a stochastic blockmodeling application and a discrete optimization solution procedure for fitting the blockmodel was deemed appropriate. Our multiobjective programming approach would be viable for such an application. The key here is that the feasibility of the approach is not necessarily contingent on the deterministic vs. stochastic distinction, but rather whether the model fitting process is implemented using discrete (combinatorial optimization) or continuous (smooth) optimization procedures.

Acknowledgements

We are grateful to the Editor, Associate Editor, and three anonymous reviewers for their thoughtful and constructive comments, which led to a substantially improved paper. Douglas Steinley's work was supported in part by NIAAA Grant 5K25AA017456.

References

Airoldi, E.M., Blei, D.M., Fienberg, S.E., & Xing, E.P. (2008). Mixed-membership stochastic blockmodels. *Journal of Machine Learning Research*, *9*, 1981–2014.

Batagelj, V., Ferligoj, A., & Doreian, P. (1992). Direct and indirect methods for structural equivalence. *Social Networks*, *14*, 63–90.

Bickel, P.J., & Chen, A. (2009). A nonparametric view of network models and Newman–Girvan and other modularities. *Proceedings of the National Academy of Sciences of the United States of America*, *106*, 21068–21073.

Breiger, R.L., Boorman, S.A., & Arabie, P. (1975). An algorithm for clustering relational data with applications to social network analysis and comparison to multidimensional scaling. *Journal of Mathematical Psychology*, *12*, 328–383.

Brusco, M.J., Cradit, J.D., & Stahl, S. (2002). A simulated annealing heuristic for a bicriterion problem in market segmentation. *Journal of Marketing Research*, *39*, 99–109.

Brusco, M.J., Cradit, J.D., & Tashchian, A. (2003). Multicriterion clusterwise regression for joint segmentation settings: An application to customer value. *Journal of Marketing Research*, *40*, 225–234.

Brusco, M., Doreian, P., Mrvar, A., & Steinley, D. (2011). Linking theory, models, and data to understand social network phenomena: Two algorithms for relaxed structural balance partitioning. *Sociological Methods & Research*, *40*, 57–87.

Brusco, M., Doreian, P., Mrvar, A., & Steinley, D. (in press). An exact algorithm for blockmodeling of two-mode network data. *The Journal of Mathematical Sociology*.

Brusco, M.J., & Stahl, S. (2001). An interactive multiobjective programming approach to combinatorial data analysis. *Psychometrika*, *66*, 5–24.

Brusco, M.J., & Stahl, S. (2005). *Branch-and-bound applications in combinatorial data analysis*. New York: Springer.

Brusco, M., & Steinley, D. (2007). A variable neighborhood search method for generalized blockmodeling of two-mode binary matrices. *Journal of Mathematical Psychology*, *51*, 325–338.

Brusco, M.J., & Steinley, D. (2009a). Integer programs for one- and two-mode blockmodeling based on prespecified image matrices for structural and regular equivalence. *Journal of Mathematical Psychology*, *53*, 577–585.

Brusco, M.J., & Steinley, D. (2009b). Cross validation issues in multiobjective clustering. *British Journal of Mathematical & Statistical Psychology*, *62*, 349–368.

Brusco, M., & Steinley, D. (2010). K-balance partitioning: An exact method with application to generalized structural balance and other psychological contexts. *Psychological Methods*, *15*, 145–157.

Brusco, M., & Steinley, D. (2011). A tabu search heuristic for deterministic two-mode blockmodeling. *Psychometrika*, *76*, 612–633.

Burt, R.S. (1976). Positions in networks. *Social Forces*, *55*, 93–122.

Cartwright, D., & Harary, F. (1956). Structural balance: A generalization of Heider's theory. *Psychological Review*, *63*, 277–293.

Davis, J.A. (1967). Clustering and structural balance in graphs. *Human Relations*, *20*, 181–187.

Delattre, M., & Hansen, P. (1980). Bicriterion cluster analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *2*, 277–291.

Dempster, A.P., Laird, N.M., & Rubin, D.B. (1977). Maximum likelihood from incomplete data via the E–M algorithm. *Journal of the Royal Statistical Society. Series B*, *39*, 1–38.

DeSarbo, W.S., & Grisaffe, D. (1998). Combinatorial optimization approaches to constrained market segmentation: An application to industrial market segmentation. *Marketing Letters*, *9*, 115–134.

Doreian, P. (2008). A multiple indicator approach to blockmodeling signed networks. *Social Networks*, *30*, 247–258.

Doreian, P., Batagelj, V., & Ferligoj, A. (1994). Partitioning networks based on generalized concepts of equivalence. *The Journal of Mathematical Sociology*, *19*, 1–27.

Doreian, P., Batagelj, V., & Ferligoj, A. (2004). Generalized blockmodeling of two-mode network data. *Social Networks*, *26*, 29–53.

Doreian, P., Batagelj, V., & Ferligoj, A. (2005). *Generalized blockmodeling*. Cambridge: Cambridge University Press.

Doreian, P., & Mrvar, A. (1996). A partitioning approach to structural balance. *Social Networks*, *18*, 149–168.

Doreian, P., & Mrvar, A. (2009). Partitioning signed social networks. *Social Networks*, *31*, 1–11.

Erhgott, M., & Gandibleaux, X. (2000). A survey and annotated bibliography on multiobjective combinatorial optimization. *OR Spektrum*, *22*, 425–460.

Ehrgott, M., & Wiecek, M.M. (2005). Multiobjective programming. In J. Figueira, S. Greco, & M. Ehrgott (Eds.), *Multiple criteria decision analysis: State of the art surveys* (pp. 667–722). New York: Springer.

Ferligoj, A., & Batagelj, V. (1992). Direct multicriteria clustering algorithms. *Journal of Classification*, *9*, 43–61.

Fienberg, S.E., Meyer, M.M., & Wasserman, S.S. (1985). Statistical analysis of multiple sociometric relations. *Journal of the American Statistical Association*, *80*, 51–67.

Gilks, W.R., Richardson, S., & Spieglhalter, D.J. (Eds.) (1996). *Markov chain Monte Carlo in practice*. New York: Chapman & Hall.

Glover, F. (1989). Tabu search—Part I. *ORSA Journal on Computing*, *1*, 190–206.

Glover, F. (1990). Tabu search—Part II. *ORSA Journal on Computing*, *2*, 4–32.

Glover, F., & Laguna, M. (1993). Tabu search. In C. Reeves (Ed.), *Modern heuristic techniques for combinatorial problems* (pp. 70–141). Oxford: Blackwell.

Goldenberg, A., Zheng, A., Fienberg, S.E., & Airoldi, E.M. (2009). A survey of statistical network models. *Foundations and Trends in Machine Learning*, *2*, 1–117.

Govaert, G., & Nadif, M. (2003). Clustering with block mixture models. *Pattern Recognition*, *36*, 463–473.

Handcock, M.S., Raftery, A.E., & Tantrum, J. (2007). Model-based clustering for social networks (with discussion). *Journal of the Royal Statistical Society. Series A*, *170*, 301–354.

Handl, J., Kell, D.B., & Knowles, J. (2007). Multiobjective optimization in bioinformatics and computational biology. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *4*, 279–292.

Handl, J., & Knowles, J. (2007). An evolutionary approach to multiobjective clustering. *IEEE Transactions on Evolutionary Computation*, *11*, 56–76.

Hartigan, J. (1972). Direct clustering of a data matrix. *Journal of the American Statistical Association*, *67*, 123–129.

Heider, F. (1946). Attitudes and cognitive organization. *The Journal of Psychology*, *21*, 107–112.

Holland, P.W., Laskey, K.B., & Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social Networks*, *5*, 109–137.

Karrer, B., & Newman, M.E.J. (2011). Stochastic blockmodels and community structure in networks. *Physical Review E*, *83*, 016107. doi:10.1103/PhysRevE.83.016107.

Kulturel-Konak, S., Coit, D.W., & Baheranwala, F. (2008). Pruned Pareto-optimal sets for the system redundancy allocation problem based on prioritized objectives. *Journal of Heuristics*, *14*, 335–357.

Kulturel-Konak, S., Smith, A.E., & Norman, B.A. (2006). Multi-objective tabu search using a multinomial probability mass function. *European Journal of Operational Research*, *169*, 918–931.

Lemann, T.B., & Solomon, R.L. (1952). Group characteristics as revealed in sociometric patterns and personality ratings. *Sociometry*, *15*, 7–90.

Liu, Y., Ram, S., Lusch, R., & Brusco, M. (2010). Multicriterion market segmentation: A new model, implementation and evaluation. *Marketing Science*, *29*, 880–894.

Lorrain, F., & White, H.C. (1971). Structural equivalence of individuals in social networks. *The Journal of Mathematical Sociology*, *1*, 49–80.

McCormick, W.T., Schweitzer, P.J., & White, T.W. (1972). Problem decomposition and data reorganization by a clustering technique. *Operations Research*, *20*, 993–1009.

Newcomb, T.N. (1961). *The acquaintance process*. New York: Holt, Rinehart & Winston.

Nowicki, K., & Snijders, T.A.B. (2001). Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, *96*, 1077–1087.

Ripon, K.S.N., Tsang, C.-H., Kwong, S., & Ip, M.-K. (2006). Multi-objective evolutionary clustering using variable-length real jumping genes genetic algorithm. In *Proceedings of the 18th international conference on pattern recognition* (pp. 1200–1203).

Saha, S., & Bandyopadhyay, S. (2010). A new multiobjective clustering technique based on the concepts of stability and symmetry. *Knowledge and Information Systems*, *23*, 1–27.

Sampson, S.F. (1968). *A novitiate in a period of change: An experimental case study of relationships*. Unpublished Ph.D. dissertation, Department of Sociology, Cornell University, Ithaca, NY.

Steinley, D., Brusco, M.J., & Wasserman, S. (2011). Clusterwise p^* models for social network analysis. *Statistical Analysis and Data Mining 4*, 487–496.

Ulungu, E., & Teghem, J. (1994). The two-phase method: An efficient procedure to solve biobjective combinatorial optimization problems. *Foundations of Computing and Decision Sciences*, *20*, 149–165.

Wasserman, S., & Faust, K. (1994). *Social network analysis: methods and applications*. Cambridge: Cambridge University Press.

Wasserman, S., & Pattison, P. (1996). Logit models and logistic regression for social networks: I. An introduction to Markov graphs and p^*. *Psychometrika*, *61*, 401–425.

White, H.C., Boorman, S.A., & Brieger, R.L. (1976). Social structure from multiple networks. I. Blockmodels of roles and positions. *American Journal of Sociology*, *81*, 730–779.

Žnidaršič, A., Ferligoj, A., & Doreian, P. (2012). Non-response in social networks: The impact of different non-response treatments on the stability of blockmodels. *Social Networks*, *34*, 438–450.

Manuscript Received: 23 FEB 2012
Final Version Received: 29 JUL 2012
Published Online Date: 29 DEC 2012