

RESEARCH ARTICLE

A simulated annealing heuristic for maximum correlation core/periphery partitioning of binary networks

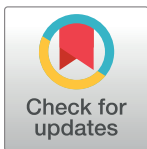
Michael Brusco^{1*}, Hannah J. Stolze², Michaela Hoffman³, Douglas Steinley³

1 Department of Marketing, Florida State University, Tallahassee, Florida, United States of America,

2 Business and Economics Department, Wheaton College, Wheaton, Illinois, United States of America,

3 Department of Psychological Sciences, University of Missouri, Columbia, Missouri, United States of America

* steinleyd@missouri.edu



OPEN ACCESS

Citation: Brusco M, Stolze HJ, Hoffman M, Steinley D (2017) A simulated annealing heuristic for maximum correlation core/periphery partitioning of binary networks. PLoS ONE 12(5): e0170448. <https://doi.org/10.1371/journal.pone.0170448>

Editor: Sergio Gómez, Universitat Rovira i Virgili, SPAIN

Received: January 15, 2016

Accepted: January 5, 2017

Published: May 9, 2017

Copyright: © 2017 Brusco et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: Data are available from Figshare <https://dx.doi.org/10.6084/m9.figshare.4046661>.

Funding: The only funding to support this work was the NIH grant mentioned – R21AA022074, awarded to Douglas Steinley. The funder had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Abstract

A popular objective criterion for partitioning a set of actors into core and periphery subsets is the maximization of the correlation between an ideal and observed structure associated with intra-core and intra-periphery ties. The resulting optimization problem has commonly been tackled using heuristic procedures such as relocation algorithms, genetic algorithms, and simulated annealing. In this paper, we present a computationally efficient simulated annealing algorithm for maximum correlation core/periphery partitioning of binary networks. The algorithm is evaluated using simulated networks consisting of up to 2000 actors and spanning a variety of densities for the intra-core, intra-periphery, and inter-core-periphery components of the network. Core/periphery analyses of problem solving, trust, and information sharing networks for the frontline employees and managers of a consumer packaged goods manufacturer are provided to illustrate the use of the model.

Introduction

The problem of partitioning a set of actors into two clusters, core and periphery, is a well-studied problem in the social network literature [1–8]. Excellent recent reviews of core/periphery analysis are available in [9, 10]. Conceptually, the goal is to partition the actors so as to maximize the intra-core density, while minimizing the intra-periphery density. This goal can be operationalized in different ways. One approach employs a continuous model for developing a core-to-periphery continuum whereby a degree “coreness” is established for each actor [3, 5]. However, the continuous approach does not explicitly assign actors to core and periphery subsets. Contrastingly, the discrete core/periphery approaches explicitly allocate the actors into mutually exclusive and exhaustive subsets. Throughout the remainder of this paper, attention is restricted to discrete core/periphery methods.

There are a variety of criterion functions that can be used for discrete core/periphery partitioning. Examples include Hamming distances [2, 11] and counts of inconsistencies relative to

the ideal structure [4, 12, 13]. One criterion that has considerable appeal is the maximization of the correlation between an ideal and observed structure associated with intra-core and intra-periphery ties. This criterion, which is the focus of our paper, was originally proposed in [1] and is incorporated in the Ucinet software system [11]. The heuristic procedure designed in [1] for obtaining solutions to the maximum correlation core/periphery partitioning problem is a variation of the genetic algorithm [14].

The study [2] compared four approaches for the maximum correlation core/periphery partitioning problem: (i) the Ucinet genetic algorithm [1, 9], (ii) a genetic algorithm implementation available as a built-in function in *Mathematica*, (iii) a simulated annealing algorithm [15] available as a built-in function in *Mathematica*, and (iv) a *Mathematica* coding of a relocation/exchange algorithm [16]. The principal conclusions in [2] were that the *Mathematica* implementations of the genetic algorithm and Kernigan-Lin heuristic were more robust than the Ucinet genetic algorithm and simulated annealing; however, these comparisons were limited to small networks with $n = 20$ or fewer actors.

It is important to establish from the outset that our goal is not to tout maximum correlation core/periphery partitioning as the best approach for analyzing core/periphery structure. Indeed, there are many other approaches available in the literature that are better-suited for certain applications. Nevertheless, maximum correlation core/periphery partitioning does have a history in the literature and is available in a popular software program. Accordingly, the purpose of our paper is to show that, contrary to earlier findings [2], an efficient implementation of simulated annealing can be a viable approach to this problem. The process for accomplishing this purpose is twofold. First we present a formal statement of the maximum correlation core/periphery partitioning problem originally proposed in [1] and also studied in [2]. An important aspect of this presentation is the efficient computation of the correlation between two binary vectors [17] and the process for capitalizing on this efficiency in a heuristic implementation. Although the precise computational details of prior implementations is not entirely clear, there is some evidence that previous approaches to calculating the correlation measure have been somewhat inefficient. For example, in the study reported in ([2], p. 176), it was noted that trial solutions were evaluated by “. . . recomputing the fitness from scratch and subtracting it from the old fitness.” The method for computing correlation between binary vectors that is described herein is appreciably more efficient, and this is a profound advantage when tackling large instances of the maximum correlation core/periphery partitioning problem.

Second, a simulated annealing heuristic using a reciprocal cooling scheme is proposed, which requires the selection of only one parameter: the number of iterations. Although the results in [2] indicated that the *Mathematica* implementation of simulated annealing performed poorly, we believe that our simulated annealing approach using reciprocal cooling is much more effective. To demonstrate this effectiveness, we ensure that, for modestly-sized empirical networks from the literature, the proposed algorithm consistently produces the same criterion function value across multiple restarts. Moreover, the simulated annealing algorithm is evaluated across a broad range of synthetic networks that are much larger than those considered in previous studies [2–5]. Specifically, the synthetic networks range in size from 500 to 2000 actors and have different levels of intra-core, intra-periphery, and inter-core-periphery densities. The proposed algorithm is used to conduct core/periphery analyses for problem solving, trust, and information sharing networks corresponding to frontline employees of a consumer packaged goods manufacturer. Finally, the simulated annealing algorithm is applied to a larger real-world network and its performance is compared to results obtained using Ucinet [11].

The next section presents the formal statement of the maximum correlation core/periphery partitioning problem that is studied in this paper. This is followed by a description of the proposed simulated annealing heuristic. The new heuristic is then applied to several small empirical networks. Subsequently, a simulation study is offered to evaluate the computational efficiency of the simulated annealing heuristic under various data conditions. The heuristic is then applied to three networks associated with employees from a consumer packaged goods manufacturer. The paper concludes with a brief summary, along with limitations and extensions for future research.

Maximum correlation core/periphery partitioning

Model formulation

The simulated annealing algorithm for the maximum correlation core/periphery partitioning of binary networks has sufficient flexibility to handle either symmetric or asymmetric network matrices. The formulation of the underlying optimization problem assumes that n is the number of actors in the network and that $T = \{1, 2, \dots, n\}$ is the set of indices for those actors. An $n \times n$ binary matrix $\mathbf{A} = [a_{ij}]$, of network ties among the actors is assumed to be available, where $a_{ij} = 1$ if there is a tie from actor i to actor j and $a_{ij} = 0$ otherwise, for $1 \leq i \neq j \leq n$. As is common in most network applications [18], the main diagonal of \mathbf{A} is ignored in the analyses. The term Π is used to denote the set of all partitions of the n actors into core and periphery subsets, and $\pi = \{T_1, T_0\}$ is a partition from Π , where T_1 is the set of actors assigned to the core and T_0 is the set of actors assigned to the periphery.

The number of actors in core and periphery of partition π are denoted as $n_C = |T_1|$ and $n_P = |T_0|$, respectively. Likewise, the number of intra-core and intra-periphery dyads for partition π are defined as $d_C(\pi) = n_C^2 - n_C$ and $d_P(\pi) = n_P^2 - n_P$, respectively. The total number of intra-core and intra-periphery dyads for partition π is $d(\pi) = d_C(\pi) + d_P(\pi)$. The total number of intra-core violations for partition π (zeros in core submatrix) is:

$$v_C(\pi) = \sum_{\{i \neq j\} \in T_1} (1 - a_{ij}). \tag{1}$$

The total number of intra-periphery violations (ones in periphery submatrix) for partition π is:

$$v_P(\pi) = \sum_{\{i \neq j\} \in T_0} a_{ij}. \tag{2}$$

We also defined $\mathbf{x}(\pi)$ as a vector corresponding to an ideal core/periphery structure for partition π , where the first $n_C^2 - n_C$ elements of the vector are ones (i.e., a perfectly connected core) and the last $n_P^2 - n_P$ elements of the vector are zeros (i.e., a perfectly unconnected periphery). Similarly, the vector $\mathbf{y}(\pi)$ corresponds to the observed core/periphery structure for partition π , where the first $n_C^2 - n_C$ elements are the values of a_{ij} among all pairs of actors $\{i \neq j\} \in T_1$ and the last $n_P^2 - n_P$ elements are the values of a_{ij} among all pairs of actors $\{i \neq j\} \in T_0$. The term $r_{XY}(\pi)$ represents the bivariate correlation between the vectors $\mathbf{x}(\Pi)$ and $\mathbf{y}(\pi)$. The optimization problem for the maximum correlation core/periphery partitioning of binary networks as proposed in [1] is:

$$\text{Maximize : } r_{XY}(\pi) \tag{3}$$

$$\text{Subject to : } \pi \in \Pi. \tag{4}$$

Succinctly, Eq (4) ensures that a partition (π) is selected from the set of all partitions (Π) with the goal of maximizing the correlation criterion function (3).

Efficient computation of $r_{XY}(\pi)$ and fast updating

Given the substantial length of the \mathbf{x} and \mathbf{y} vectors, it would be computationally demanding to compute $r_{XY}(\pi)$ from scratch (e.g., using common formulas for the Pearson correlation coefficient) each time a new partition was constructed in a heuristic algorithm. Fortunately, the correlation between the two binary vectors \mathbf{x} and \mathbf{y} can be computed efficiently using the constants defined above. This is important because it allows for rapid recomputation of $r_{XY}(\pi)$ using simple updates to these constants each time the partition is modified by moving an actor from the core to the periphery or vice versa. The formulas for computing the correlation are as follows:

The variance of \mathbf{x} for partition π is computed as:

$$s_x^2(\pi) = \left(\frac{d_c(\pi)}{d(\pi)}\right) \left(\frac{d_p(\pi)}{d(\pi)}\right). \tag{5}$$

The variance of \mathbf{y} for partition π is computed as:

$$s_y^2(\pi) = \left(\frac{(d_c(\pi) - v_c(\pi) + v_p(\pi))}{d(\pi)}\right) \left(\frac{(d_p(\pi) - v_p(\pi) + v_c(\pi))}{d(\pi)}\right). \tag{6}$$

The covariance between \mathbf{x} and \mathbf{y} for partition π is computed as:

$$s_{XY}(\pi) = \left(\frac{(d_c(\pi) - v_c(\pi))}{d(\pi)}\right) - \left(\frac{d_c(\pi)}{d(\pi)}\right) \left(\frac{(d_c(\pi) - v_c(\pi) + v_p(\pi))}{d(\pi)}\right). \tag{7}$$

The correlation between \mathbf{x} and \mathbf{y} for partition π is computed as:

$$r_{XY}(\pi) = \left(\frac{s_{XY}}{\sqrt{s_x^2 s_y^2}}\right). \tag{8}$$

As an illustration of the formulae for computing the correlation measure, we consider an asymmetric core/periphery solution originally reported in [1], and later confirmed in [2]. The binary asymmetric network matrix corresponds to co-citation ties among $n = 20$ social work journals [19]. The core/periphery partition, π , reported in ([1], p. 385) consists of $n_c(\pi) = 6$ core journals and $n_p(\pi) = 14$ periphery journals. Applying the formulas in the previous subsection yields $d_c(\pi) = 6^2 - 6 = 30$, $d_p(\pi) = 14^2 - 14 = 182$, and $d(\pi) = 30 + 182 = 212$. From the table published in ([1], p. 385), it is straightforward to count $v_c(\pi) = 2$ intra-core violations (zeros in the intra-core submatrix) and $v_p(\pi) = 8$ intra-periphery violations (ones in the intra-periphery submatrix—ignoring the main diagonal). From Eq (5), the variance for the ideal structure is $s_x^2(\pi) = \left(\frac{30}{212}\right) \left(\frac{182}{212}\right) = .121485$. Likewise, from Eq (6), the variance for the observed structure is $s_y^2(\pi) = \left(\frac{(30-2+8)}{212}\right) \left(\frac{(182-8+2)}{212}\right) = .140975$. The covariance is computed as $s_{XY}(\pi) = \left(\frac{30}{212}\right) - \left(\frac{30}{212}\right) \left(\frac{(30-2+8)}{212}\right) = .108046$ using Eq (7). Finally, the correlation is computed using Eq (8) as $r_{XY}(\pi) = \left(\frac{.108046}{\sqrt{(.121485)(.140975)}}\right) = .82561$. When rounded to three decimal places, this correlation value comports to the one reported in ([1], p. 385) and is identical to the five-decimal place value reported in ([2], p. 171).

The proposed computational approach is particularly valuable for rapidly assessing the impact of neighborhood search operations applied to a given partition. To describe the process for updating the correlation measure, assume that a new partition π' is constructed by moving actor h from the core (T_1) to the periphery (T_0). This move results in $n_c(\pi') = n_c(\pi) - 1$ and $n_p(\pi') = n_p(\pi) + 1$. The formulas in the previous subsection can then be directly applied to

obtain $d_C(\pi')$ and $d_P(\pi')$, respectively. An update of the intra-core violations is provided as follows:

$$v_C(\pi') = v_C(\pi) - \sum_{\{j \neq h\} \in T_1} (1 - a_{hj}) + (1 - a_{jh}). \tag{9}$$

Essentially, Eq (9) removes the contribution to intra-core violations that stem from actor h 's relationship to all other actors in the core, which is appropriate given h 's removal from the core. Likewise, because h is moved into the periphery, it is necessary to increase the intra-periphery violations (ones in the periphery submatrix) that might arise from h 's inclusion in the periphery. This is accomplished as follows:

$$v_P(\pi') = v_P(\pi) + \sum_{\{j \neq h\} \in T_0} (a_{hj} + a_{jh}). \tag{10}$$

After making these rapid updates, the correlation for the trial partition π' is readily computed. In the case where the new partition π' is realized by moving actor h from the periphery (T_0) to the core (T_1), the result is $n_C(\pi') = n_C(\pi) + 1$ and $n_P(\pi') = n_P(\pi) - 1$. The values of $d_C(\pi')$ and $d_P(\pi')$ would be updated accordingly. An update of the intra-core violations is provided as follows:

$$v_C(\pi') = v_C(\pi) + \sum_{\{j \neq h\} \in T_1} ((1 - a_{hj}) + (1 - a_{jh})). \tag{11}$$

Eq (11) augments the contribution to intra-core violations that stem from actor h 's inclusion in the core via examination of its ties to all other actors in the core. In a similar manner, because h is removed from the periphery, it is necessary to decrease the intra-periphery violations (ones in the periphery submatrix) in accordance with h 's ties to other actors in the periphery. This is accomplished as follows:

$$v_P(\pi') = v_P(\pi) - \sum_{\{j \neq h\} \in T_0} (a_{hj} + a_{jh}). \tag{12}$$

Methods

Existing procedures

Solution procedures for the maximum correlation core/periphery partitioning problem can be divided into two categories: (i) exact and (ii) heuristic. Exact methods are assured to produce a globally optimal solution to the problem posed by Eqs 6 and 7, whereas heuristic methods do not provide assurance. One exact solution approach is to explicitly generate all partitions in Π and select the one that provides the largest value for $r_{XY}(\pi)$. The feasibility of this approach is likely restricted to networks with 30 or fewer actors, as the number of partitions in Π exceeds one billion when $n = 30$. An alternative exact approach is to employ branch-and-bound methods [4]. Unfortunately, the efficient design of strong bounds for the maximum correlation criterion function is not trivial and, even if good bounds could be established, the size of networks that could be tackled would remain modest.

Heuristic procedures for the maximum correlation core/periphery problem include the genetic algorithm available in Ucinet [1, 11], as well the exchange algorithm, simulated annealing, and genetic algorithm approaches considered in [2]. Other metaheuristics such as tabu search [20], variable neighborhood search [21] would also be viable. In light of the fact that the built-in simulated annealing program in *Mathematica* performed so poorly in the study completed in [2], we sought to design an effective simulated annealing implementation for the maximum correlation core/periphery partitioning problem. This method is described in the next subsection

Proposed simulated annealing heuristic

Simulated annealing was independently developed in [22] and [23] as a heuristic procedure for combinatorial optimization problems. The method is based on an analogy to the metallurgical process of annealing in statistical mechanics [24], whereby a better final energy state is often achieved by periodically re-heating the metal during the cooling process. The adaptation of this concept for optimization problems is that, during a local-search process, periodically accepting a trial solution (π') that has a criterion function value that is worse than that of the incumbent solution (π) will ultimately lead to the achievement of a better final criterion function value. The probability of accepting a solution that worsens the criterion function is controlled by the extent of the worsening, which in our context is $r_{XY}(\pi') - r_{XY}(\pi) < 0$, as well as the current temperature (τ) of the system. During the execution of the algorithm, the value of τ is systematically reduced to lower the probability that inferior solutions will be accepted. The process for reducing τ is known as the cooling scheme.

There are a variety of possible cooling schemes that can be used for simulated annealing, including the linear, exponential (or geometric), reciprocal, and logarithmic approaches (see [25–27]). In this paper, we use the reciprocal cooling scheme. Given an initial temperature, $\tau(1)$, and an upper limit on the number of trial solutions generated, ξ , the temperature, $\tau(q)$, for trial solution q is computed as $\tau(q) = \tau(1)/q$ using the reciprocal cooling scheme.

A natural choice for the initial temperature is $\tau(1) = 1$ because the maximum possible value of the correlation measure is one. The relocation of an actor from the core to the periphery (or vice versa) will typically produce a very small change (much less than one) in the criterion function, particularly for larger networks. Therefore, in the early stages of the algorithm with a larger cooling parameter, almost every trial solution will be accepted regardless of whether it improves or worsens the criterion function value. However, by trial solution $q = 10,000$, the value of $\tau(10,000) = 0.0001$, and the probability of accepting a solution that worsens the criterion function only slightly is quite small.

If the initial temperature is fixed at $\tau(1) = 1$, then the only user-specified parameter for the algorithm is ξ . If ξ is chosen too small, then the algorithm might not reach temperatures that are low enough to ensure convergence to a good solution. On the other hand, choosing a value of ξ that is too large can be computationally wasteful, as no inferior trial solutions will be accepted once the temperature is too small. We have found that a range of $100,000 \leq \xi \leq 1,000,000$ trial solutions seems to generally perform well for a diverse range of network sizes.

Fig 1 provides a pseudo code for our implementation of simulated annealing with a reciprocal cooling scheme. An initial partition, π , is randomly generated by assigning each actor to either the core or the periphery based on a uniform distribution with 50% probability for each subset. This initial partition is then installed as the best-found partition, π^* . The algorithm then generates the selected number of trial solutions (ξ). For each trial solution, an actor is randomly selected, and the cluster membership of that actor is changed from core to periphery or periphery to core as appropriate. The computation of $r_{XY}(\pi')$ for trial solution π' is performed using the Eqs 8–13. If $r_{XY}(\pi') \geq r_{XY}(\pi)$, then π' replaces π as the incumbent solution and a check is also made to see if this new incumbent has a greater correlation index and should therefore replace the current best-found solution π^* . If $r_{XY}(\pi') < r_{XY}(\pi)$, then π' replaces π as the incumbent solution with a probability of $\exp\left(\frac{r_{XY}(\pi') - r_{XY}(\pi)}{\tau(q)}\right)$.

Application to empirical matrices

Selected networks

Our principal focus in this section is to ensure that the simulated annealing algorithm will consistently provide good solutions for modestly-sized networks. Four networks were selected

```

GENERATE an initial partition,  $\pi$ , and install it as the best-found partition,  $\pi^* = \pi$ .
SET PARAMETERS:
    Specify the initial temperature,  $\tau(1) = 1$ 
    Specify the number of iterations,  $\xi$ 
For  $q = 1$  to  $\xi$ 
    Randomly select an actor,  $h$ 
    If  $h \in T_1$ , then
         $\pi' = \{T_1 \setminus \{h\}, T_0 \cup \{h\}\}$ 
        Use Equations 12-13 to compute  $r_{XY}(\pi')$ 
    Else
         $\pi' = \{T_1 \cup \{h\}, T_0 \setminus \{h\}\}$ 
        Use Equations 14-15 to compute  $r_{XY}(\pi')$ 
    End if
    If  $r_{XY}(\pi') > r_{XY}(\pi)$  then
        set  $\pi = \pi'$ 
        If  $r_{XY}(\pi) > r_{XY}(\pi^*)$  then
            set  $\pi^* = \pi$ 
        End if
    Else
        draw a uniform random number,  $rnd$ , on  $[0,1]$ 
        If  $rnd < \exp\left(\frac{(r_{XY}(\pi') - r_{XY}(\pi))}{\tau(q)}\right)$  then
            set  $\pi = \pi'$ 
        End if
    End if
    Set  $\tau(q+1) = \tau(1)/(q+1)$ 
Next  $q$ 
RETURN  $\pi^*$  as the partition.

```

Fig 1. Simulated annealing (reciprocal cooling) pseudo code.

<https://doi.org/10.1371/journal.pone.0170448.g001>

from the literature for analysis. The first two networks correspond to co-citation network data for 20 social work journals during the period from 1985 to 1986 [19]. These network data have been explored in several previous studies [1–5, 11]. The first network matrix corresponds to a symmetric dichotomization ([1], p. 382) of the social work journal co-citation data, whereas the second is the asymmetric dichotomized data ([1], p. 385).

The third network matrix corresponds to the Kansas Search and Air Rescue Communication (Kansas SAR) study that was originally reported in [28]. The asymmetric dichotomized matrix was adapted from the one published in ([13], p. 55) and reflects message sending and receiving ties among the $n = 20$ network agencies. The fourth matrix is a 28×28 asymmetric dichotomized co-citation matrix among statistical and quantitative psychology (and other social science) journals. This matrix was taken from a study in [4], which adapted the network from co-citation data published in ([29], p. 547).

Analysis and results

The simulated annealing algorithm was programmed in Fortran 90 and implemented on a desktop computer using an Intel® Core™ i7-4790 CPU @ 3.6GHz with 8 GB of RAM. The algorithm was applied to each of the four empirical network matrices using $\xi = 100,000$ as the number of trial solutions. For each network matrix, implementation of the algorithm was repeated 10 times using a different random initial partition for each repetition. The primary performance measure was the *attraction rate*, which represents the number of repetitions for which the best criterion function value was obtained. The secondary performance measure was the maximum computation time across all 10 restarts. The results are reported in Table 1.

The results in Table 1 indicate that, for each of the four network matrices, the simulated annealing algorithm produced the same criterion function value for all 10 repetitions. This consistency is important because it shows that, for modestly-sized networks, the algorithm is not highly sensitive to the initial starting partitions. The maximum computation times reported in Table 1 are also modest. Across all matrices and repetitions, the maximum computation time across the 10 restarts was only .01 seconds.

For the two social work co-citation networks, the partitions obtained correspond to those originally published in ([1], pp. 382, 385) and later confirmed in [2]. For the psychological/statistical network, the core was very similar to the one obtained and displayed in ([4], p. 15, Table 2) using a criterion function that summed the number of (non-diagonal) zeros in the core and the number of (non-diagonal) ones in the periphery. The core from the partition reported in [4] consisted of nine journals: {*Psychometrika*, *British Journal of Mathematical and Statistical Psychology*, *Multivariate Behavioral Research*, *Psychological Bulletin*, *Annals of Statistics*, *Biometrics*, *Biometrika*, *Journal of the American Statistical Association*, *Journal of the Royal Statistical Society B*}. The first four members of the core are statistical psychology journals and last five are elite statistical outlets. The partition we obtained using the maximum correlation criterion function added the journal *Applied Statistics* to the core. The practical merit of the inclusion of this journal in the core is debatable, as it is more applied in nature relative to the other five statistical journals.

The core/periphery partition for the Kansas SAR network is displayed in Fig 2. The value of the criterion function for this partition is $r_{XY}(\pi) = .71047$. The naming convention for the

Table 1. Attraction rates and computation times for empirical network matrices.

Network	Size	$r_{XY}(\pi)$	Attraction rate	Maximum computation time
Social work journal co-citations				
Symmetric, dichotomized	$n = 20$.85959	10 out of 10	.01
Sources: [1, 17]				
Social work journal co-citations				
Asymmetric, dichotomized	$n = 20$.82561	10 out of 10	.01
Sources: [1, 17]				
Statistics/Psychology co-citations				
Asymmetric, dichotomized	$n = 28$.50694	10 out of 10	.01
Sources: [4, 27]				
Kansas SAR Communications				
Asymmetric, dichotomized	$n = 20$.71047	10 out of 10	.01
Sources: [11, 26]				

Note—'Attraction Rate' is the number of repetitions (out of 10) for which the reported criterion value was achieved. The maximum computation times are in seconds.

<https://doi.org/10.1371/journal.pone.0170448.t001>

agencies in the network was provided in ([11], p. 54). The core consists of nine agencies, most of which exhibit a high degree of symmetry with respect to their sending and receiving of messages. The notable exceptions are ‘Shawney’ and ‘Burl Police’, which send messages to most of the other core members, but do not receive messages from the other core members.

A Simulation study

Simulation design. Simulated annealing proved effective for the modestly-sized networks in the previous section. However, our goal here is to provide a more robust simulation-based evaluation of the method across a set of much larger networks with various core sizes and density characteristics. Five data features were manipulated in the experimental design. The first feature, the number of actors, was tested at three levels: $n = 500$, 1000 , and 2000 . The second feature controlled the relative sizes of the core and periphery subsets, and was tested at three levels: (a) the core size is $.05n$ (i.e., 5% of the actors in the core), (b) the core size is $.10n$, and (c) the core size is $.20n$. We selected these levels because the core is typically smaller than the periphery. The third, fourth, and fifth design features corresponded to the intra-core density, intra-periphery density, and inter-core-periphery density, respectively. The levels of the third design feature were 80% and 70%. The levels of the fourth design were 30% and 20%. The levels of the fifth design feature were 40% and 60%.

A fully-crossed design was employed, resulting in $3 \times 3 \times 2 \times 2 \times 2 = 72$ cells. Three problem replications were generated in each cell, which yielded a total of 216 test problems in the experiment. The simulated annealing heuristic was applied to each of the 216 test problems for three different numbers of trial solutions: (i) $\xi = 100,000$, (ii) $\xi = 500,000$, and (iii) $\xi = 1,000,000$. The criterion function values and computation times were collected for each of the three runs of the simulated annealing heuristic for each test problem.

The results for $\xi = 1,000,000$ trial solutions provides a benchmark for evaluating the results at the other two levels of ξ . Because the results for the simulated algorithm cannot worsen as ξ is increased, the best-found correlation criterion function value for each test problem is

Table 2. Summary of simulation results.

Feature	Level	MRCD Results		MPCD Results		MPBF Results	
		100,000	500,000	100,000	500,000	100,000	500,000
Number of actors	$n = 500$.00003	.00000	.02947	.00053	25.00	94.44
	$n = 1000$.00028	.00001	.25364	.00432	13.89	48.61
	$n = 2000$.00147	.00003	1.47529	.03155	11.11	20.83
Percentage of actors in core	5%	.00101	.00002	1.24427	.02427	1.39	40.28
	10%	.00067	.00002	.47884	.01090	1.39	44.44
	20%	.00009	.00000	.03529	.00123	47.22	79.17
Intra-core density	70%	.00062	.00001	.60961	.01304	12.04	51.85
	80%	.00057	.00001	.56266	.01122	21.30	57.41
Intra-periphery density	20%	.00057	.00001	.45406	.00947	20.37	57.41
	30%	.00061	.00001	.71821	.01480	12.96	51.85
Inter core-periphery density	40%	.00054	.00001	.70895	.01239	25.93	62.96
	60%	.00065	.00002	.46332	.01188	7.41	46.30
Overall		.00059	.00001	.58613	.01213	16.67	54.63

Note—the results in the table correspond to 100,000 and 500,000 temperature reductions relative to 1 million:

MRCD—mean raw correlation differences

MPCD—mean percentage correlation differences

MPBF—mean percentage of best-found correlation values

<https://doi.org/10.1371/journal.pone.0170448.t002>

	1	3	5	6	7	9	14	15	17	2	4	8	10	11	12	13	16	18	19	20
1 OSherriff	x	1	1	1	1	1	0	1	1	1	0	0	0	1	0	0	1	0	0	1
3 Coroner	1	x	1	1	0	1	0	0	1	0	1	0	0	1	0	0	0	0	0	0
5 Highway P	1	1	x	1	1	0	1	0	1	0	1	0	0	1	1	1	1	0	0	1
6 Parks n Res	1	1	1	x	1	1	1	0	0	0	1	0	0	1	1	0	0	0	0	0
7 Game n Fish	1	1	1	1	x	1	0	0	0	0	1	0	1	0	0	0	1	0	0	0
9 Army Corps	1	1	1	1	1	x	0	0	1	0	0	0	0	1	0	0	0	0	0	0
14 Shawney	1	1	1	1	1	1	x	0	1	0	0	0	0	0	1	1	0	1	0	1
15 Burl Police	1	1	1	1	1	1	1	x	1	0	1	0	1	1	1	1	1	1	0	1
17 Red Cross	1	1	1	1	1	1	1	0	x	1	0	0	1	0	0	0	0	0	0	0
2 Civil Def	1	1	1	1	1	1	1	0	1	x	0	0	0	0	0	0	1	0	0	0
4 Attorney	1	1	1	1	1	1	1	0	0	1	x	0	0	0	0	0	0	0	0	0
8 Kansas DOT	1	0	1	0	0	0	0	0	0	1	0	x	1	0	0	0	0	0	0	0
10 Army Reserve	1	0	1	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0
11 Crable Amb	1	1	1	1	1	1	0	0	0	0	0	0	0	x	1	0	0	0	0	0
12 Frank Co. Amb	1	0	1	0	1	0	0	0	1	0	0	0	0	0	x	0	0	0	0	0
13 Lee Rescue	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	x	0	0	0	0
16 Lynd Police	1	0	1	1	1	0	0	0	0	1	0	0	0	1	0	0	x	0	0	0
18 Topica FD	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	x	0	0
19 Carb FD	1	0	1	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	x	0
20 Topeka RBW	1	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	x

Fig 2. The core/periphery partition for the Kansas SAR network obtained using the simulated annealing algorithm. The naming system for the agencies is from [11, p. 54), and the rows are the 'sending' agencies, whereas the columns are the 'receiving' agencies. The two boxes contain the intra-core and intra-periphery elements.

<https://doi.org/10.1371/journal.pone.0170448.g002>

realized for $\xi = 1,000,000$. For the settings of $\xi = 100,000$ and $\xi = 500,000$, two performance measures are computed for each test problem: (i) the raw deviation between the measured correlation value and best-found correlation value for $\xi = 1,000,000$, and (ii) the raw deviation measured expressed as a percentage of the best-found correlation value for $\xi = 1,000,000$. From these performance measures, three summary measures are computed, both overall and for different levels of the design features: (i) mean raw correlation deviation (MRCD), (ii) mean percentage correlation deviation (MPCD), and (iii) mean percentage of best-found correlation values (MPBF).

Simulation results

The computation times for the simulated annealing algorithm were affected by two conditions: the number of actors (n) and the number of trial solutions (ξ). Computation times ranged

from roughly one-half second for the test problems with $n = 500$ and $\xi = 100,000$, to roughly 30 seconds for test problems with $n = 2000$ and $\xi = 1,000,000$. These times are modest and suggest that the algorithm is computationally feasible for a sufficiently large number of trial solutions (e.g. $\xi \geq 1,000,000$) for networks with several thousand vertices. Nevertheless, it should be noted that computation time is not a linear function of n . For each value of ξ , the average computation time for the $n = 1000$ test problems is slightly more than *double* the average of the $n = 500$ test problems. However, the average computation time for the $n = 2000$ test problems is slightly more than *triple* the average of the $n = 1000$ test problems.

Table 2 reports the MRCD, MPCD, and MPBF values for each level of each design feature of the simulation experiment. The overall (across all test problems) averages are also reported at the bottom of the table. The overall MRCD values for $\xi = 100,000$ and $\xi = 500,000$ were .00059 and .00001, respectively. Moreover, across all test problems, the maximum difference between the correlation criterion function measured for $\xi = 100,000$ and $\xi = 1,000,000$ was only .00289 (.04981 vs. .05270). Likewise, across all problems, the maximum difference between the correlation criterion function measured for $\xi = 500,000$ and $\xi = 1,000,000$ was only .00011 (.13111 vs. .13122). The overall MPCD values for $\xi = 100,000$ and $\xi = 500,000$ were .58613% and .01213%, respectively. Across, all test problems, the maximum percentage deviations were 5.48387% and .11021% for $\xi = 100,000$ and $\xi = 500,000$, respectively. Finally, the overall MPBF values for $\xi = 100,000$ and $\xi = 500,000$ were 16.67% and 54.63%, respectively.

The intra-core density, intra-periphery density, and inter-core-periphery density design features exhibited a relatively modest effect on the MRCD, MPCD, and MPBF measures. However, these measures were more strongly affected by the levels for the number of actors and the percentage of actors in the core. Consider, for example, at $n = 500$, the implementations using $\xi = 100,000$ and $\xi = 500,000$ matched the best-found correlation values for 25.00% and 94.44% of the test problems, respectively. However, for $n = 2000$, the $\xi = 100,000$ and $\xi = 500,000$ implementations matched the best-found correlation values for $\xi = 1,000,000$ for only 11.11% and 20.83% of the test problems, respectively.

The $\xi = 100,000$ and $\xi = 500,000$ implementations performed reasonably well when the percentage of actors in the core was 20%, matching the best-found correlation values for 47.22% and 79.17% of the test problems, respectively. However, their attraction to the best-found solution was markedly less impressive for the smaller core size levels of 5% and 10%. For these smaller percentages of actors in the core, the $\xi = 100,000$ implementation only matched the best-found correlation value about 1% of the time, whereas the $\xi = 500,000$ implementation matched the best-found correlation roughly 40% of the time.

Core/Periphery analyses of organizational networks

Network relations

Three networks were used to demonstrate the simulated annealing heuristic for the maximum correlation core/periphery partitioning problem. The networks were obtained using the full roster method [18] for $n = 25$ managers and frontline employees of a consumer packaged goods manufacturer. The 25-member bounded social group consisted of frontline truck drivers and managers assigned to a specific regional cross-dock warehouse location. One-hundred percent of the bounded social group responded to the survey, ranking each of the other members of their social network for the most rigorous recall-based elicitation of social network relations and ultimately, structure [30]. The research team administered the surveys directly to the frontline employees and management team at a scheduled monthly meeting.

The three relations examined were: (i) *problem solving*—that is, identifying those from whom assistance is sought when a problem arises, (ii) *trust*—identification of those persons

from whom advice is trusted, and (iii) *information sharing*—identification of those persons from whom information about upcoming store promotions was sought. The three resulting network matrices were asymmetric and, for each network, the rows of the corresponding network correspond to the tie-senders (the respondents) and the columns correspond to the tie-receivers (the alters).

Analysis and results

Ten repetitions of the simulated annealing algorithm were applied to each of the three network matrices. For each matrix, the simulated annealing algorithm obtained the same partition for each of the 10 repetitions, which is consistent with the findings for the empirical network matrices investigated above. Although these strong attraction rates are not unequivocal proof that the global maximum for the correlation criterion function has been achieved, they do tend to support the likelihood of this result. Figs 3, 4 and 5 display the core/periphery partitions for problem solving, trust, and information sharing, respectively.

The trust network (see Fig 4) is far more dense (density of .60667) than the other two networks and, therefore, not surprisingly, has the largest of the cores (14 actors). The core/periphery partition for the trust network also has a larger correlation criterion function value, $r_{XY}(\pi) = .74630$, than the other two networks. The information sharing network (see Fig 5) is the least dense (density of .22167) of the three networks and has the smallest core (8 actors). However, the core/periphery partition for the information sharing network also has a solid correlation criterion function value of $r_{XY}(\pi) = .65933$. The density of the problem solving network (density of .26000) is slightly greater than that of the information sharing network, and its core is also larger at 11 actors. However, the problem solving network is the least well-structured, as its core/periphery partition has a correlation criterion value of only $r_{XY}(\pi) = .52004$.

Six of the actors, {1, 2, 3, 12, 14, 17}, are present in the cores of the partitions for the problem solving, trust, and information sharing networks. Accordingly, these six actors occupy central positions for all three relational roles within the organization. Three of the actors, {4, 7, 21}, are in the cores for the partitions for the problem solving and trust networks, but not the information sharing networks. Actor 10 is in the core for the problem solving and information sharing networks, but not the trust network. Actor 8 is in the core for the trust and information sharing networks, but not the problem solving network. Four actors, {9, 16, 22, 24} appear in the core for the trust network only, and actor 15 is in the core for the problem solving network only. Nine actors, {5, 6, 11, 13, 18, 19, 20, 23, 25}, were in the periphery for all three networks.

Efficiency and effectiveness for a larger network

To complete our analyses, we studied the efficiency and effectiveness of the simulated annealing algorithm by applying it to a larger binary network ($n = 2114$). The data correspond to a protein interaction network of yeast [31] and are available from several sources including: <http://www3.nd.edu>. We applied the 10 restarts of the simulated annealing algorithm to these data using $\xi = 100,000$. For comparison purposes, we also conducted maximum-correlation core/periphery partitioning using Ucinet [11].

Maximum-correlation core/periphery partitioning in Ucinet was accomplished using the built-in genetic algorithm with the CORR option. The algorithm required 10 hours and 47 minutes on an Intel® Core™ i7-4720HQ CPU @ 2.6GHz laptop computer with 8 GB of RAM to produce a solution with a correlation of .07486. By contrast, each of the 10 restarts of the simulated annealing algorithm required less than 10 seconds to produce a solution, and the correlation values ranged from .07649 to .07712. In summary, each of the 10 restarts produced a better solution than the one found by Ucinet and did so in dramatically less time.

Conclusions

We have presented a new simulated annealing algorithm for maximum correlation core/periphery partitioning of binary networks. The algorithm is exceptionally fast and requires minimal parameterization. A key component of the algorithm is that it uses an efficient process for updating the Pearson correlation measure between two binary vectors as trial solutions are generated. This is in contrast to some earlier implementations whereby trial solutions were evaluated by recomputing the correlation measure from scratch [2].

	1	2	3	4	7	10	12	14	15	17	21	5	6	8	9	11	13	16	18	19	20	22	23	24	25	
1	x	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
2	1	x	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
3	1	1	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
4	0	0	0	x	1	1	1	0	0	0	0	1	1	1	1	1	0	0	1	0	0	0	0	1	0	0
7	1	1	1	0	x	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
10	1	1	1	1	1	x	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	1	0	1	1	1	x	1	0	0	0	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0
14	1	1	0	1	1	1	1	x	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
15	1	0	1	1	1	0	0	0	x	1	1	1	1	1	0	0	0	1	0	1	1	1	1	1	1	0
17	1	1	1	0	1	1	0	0	0	x	1	0	0	1	1	0	0	1	1	1	1	1	1	1	1	0
21	0	0	1	0	1	0	0	0	1	1	x	0	0	0	1	0	0	1	1	1	0	1	1	1	1	0
5	0	1	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	1	0	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	0	1	0	1	0	0	0	0	0	0	x	0	0	0	0	0	0	0	0	0	0	0	0
9	0	1	1	0	0	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	0	0	0	0
11	0	1	1	0	1	1	1	0	0	0	0	0	1	0	0	x	0	0	0	0	0	0	0	0	0	0
13	0	1	1	1	1	0	0	0	0	0	0	1	1	0	0	0	x	0	0	0	0	0	0	0	0	0
16	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	0	0
18	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	x	1	0	0	0	0	0	0	0
19	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	1
20	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0
22	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	x	0	0	0	0
23	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	x	0	0	0	0
24	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	x	0	0
25	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x

Fig 3. The core/periphery partition for the problem solving network. The two boxes contain the intra-core and intra-periphery elements.

<https://doi.org/10.1371/journal.pone.0170448.g003>

The algorithm was initially applied to four small empirical networks to measure its consistency across multiple repetitions. For each of the four networks, the algorithm identified the same partition for each of its 10 repetitions. These findings suggest that the relatively poor performance of simulated annealing in the comparative analyses in [2] was an artifact of the particular generic implementation in *Mathematica*, not an indication of an inherent flaw in the simulated annealing approach itself.

Next, a simulation study was completed, wherein the algorithm was applied to much larger synthetic networks ranging in size from 500 to 2000 actors. The levels of core size, intra-core

	1	2	3	4	7	8	9	12	14	16	17	21	22	24	5	6	10	11	13	15	18	19	20	23	25
1	x	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	x	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	x	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	1	1	1	0	1	1
4	1	1	1	x	1	1	1	1	0	1	1	0	0	1	1	1	1	1	0	0	1	1	0	1	1
7	1	1	1	1	x	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	0	1	x	1	1	1	0	1	1	0	0	1	1	0	1	0	0	0	1	0	1	1
9	0	1	1	1	1	1	x	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	x	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	x	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
16	1	1	1	1	1	1	1	1	1	x	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	1	1	1	1	1	1	1	1	1	1	x	1	1	1	1	1	1	1	1	1	1	1	1	1	0
21	1	0	1	1	1	1	1	0	0	1	1	x	1	1	0	1	0	0	0	1	1	1	0	1	1
22	0	1	1	1	1	1	1	1	1	1	1	1	x	1	1	1	1	1	1	1	1	1	1	1	1
24	1	1	1	1	1	1	1	1	1	1	1	1	1	x	1	1	1	1	1	1	1	1	1	1	1
5	1	1	0	0	0	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	0	0	0
6	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	0	0
10	1	1	1	0	0	1	0	1	1	1	0	0	0	0	0	0	x	0	0	0	0	1	0	0	0
11	1	1	1	0	1	1	1	1	0	0	0	0	0	0	1	1	1	x	0	0	0	0	0	0	0
13	1	1	1	1	1	1	1	1	1	0	0	1	0	0	1	1	0	0	x	0	0	0	0	0	0
15	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0	x	0	1	1	1	0
18	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	x	0	0	0	0
19	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0	0	1
20	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	x	1	0
23	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	x	0
25	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x

Fig 4. The core/periphery partition for the trust network. The two boxes contain the intra-core and intra-periphery elements.

<https://doi.org/10.1371/journal.pone.0170448.g004>

	1	2	3	8	10	12	14	17	4	5	6	7	9	11	13	15	16	18	19	20	21	22	23	24	25	
1	x	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
2	1	x	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	x	1	1	1	0	0	0	0	1	1	0	1	0	0	0	0	0	0	1	0	0	0	1
10	1	1	1	1	x	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	1	0	1	1	x	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
14	1	1	1	1	1	1	x	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
17	1	1	1	0	0	0	0	x	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
4	0	1	0	0	0	0	0	0	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	1	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
6	0	1	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	1	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	1	1	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	1	0	0	0	0	0	0	0	0	0	1	0	x	0	0	0	0	0	0	0	0	0	0	0	0
13	0	1	0	0	0	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	0	0	0	0
15	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	x	1	0	0	0	1	0	0	0	0	0
16	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	0	0
18	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	x	1	0	0	0	0	0	0	0
19	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	x	0	0	0	0	0	0	0
20	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	x	0	0	0	0	0	0
21	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	x	0	0	0	0	0
22	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	x	0	0	0	0
23	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	x	0	0	0
24	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	x	0
25	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x

Fig 5. The core/periphery partition for the information sharing network. The two boxes contain the intra-core and intra-periphery elements.

<https://doi.org/10.1371/journal.pone.0170448.g005>

density, intra-periphery density, and inter-core-periphery density were controlled in the simulation study, and criterion function performance was evaluated at three different levels for the maximum number of iterations: 100,000, 500,000, and 1 million. The results of the simulation study indicated that 500,000 was a reasonable iteration limit even for the largest networks.

The use of the algorithm was demonstrated via an application to three networks (problem solving, trust, and information sharing) corresponding to 25 frontline employees and

managers for a consumer packaged goods manufacturer. As was the case for the four empirical network matrices from the literature, the new simulated annealing algorithm consistently provided the same maximum correlation criterion function value across 10 repetitions.

The algorithm was also applied to a much larger ($n = 2114$) protein interaction network from the literature. Ten restarts of the algorithm produced correlation values within a narrow range (.07649 to .07712). The computation time for each restart was less than 10 seconds. For comparative purposes, we tackled the same network using the maximum-correlation core/periphery partitioning capabilities in Ucinet. The genetic algorithm used by Ucinet required more than 10 hours to obtain a solution, and the correlation of .07486 for that solution was worse than that of each of the 10 restarts of simulated annealing.

There are a variety of ways in which the method in this paper can be extended. First, the analyses in this paper were restricted to the case of a single core; however, as noted in [6], there is the potential for multiple cores. Although the consideration of multiple cores increases the computational demand because of the presence of more clusters, the general computational scheme for correlation between two binary vectors is unaffected. Second, we reiterate that the computational scheme presently incorporated in the heuristic is limited to binary networks. As noted previously, the proposed heuristic was explicitly designed to capitalize on rapid computation of the correlation between binary vectors. For real-valued networks, sum-of-squares information would need to be stored for dynamic updating of the correlation measure; however, the basic structure of the simulated annealing heuristic could remain the same.

Author Contributions

Conceptualization: MB HS MH DS.

Data curation: MB.

Formal analysis: MB HS MH DS.

Funding acquisition: DS.

Investigation: MB.

Methodology: MB HS MH DS.

Project administration: MB DS.

Resources: MB DS.

Software: MB.

Supervision: MB DS.

Validation: MB HS MH DS.

Visualization: MB.

Writing – original draft: MB.

Writing – review & editing: MB HS MH DS.

References

1. Borgatti SP, Everett MG. Models of core/periphery structures. *Soc Net.* 1999; 21: 375–95.
2. Boyd JP, Fitzgerald WJ, Beck RJ. Computing core/periphery structures and permutation tests for social relations data. *Soc Net.* 2006; 28 (2): 165–78.

3. Boyd JP, Fitzgerald WJ, Mahutga MC, Smith DA. Computing continuous core/periphery structures for social relations data with MINRES/SVD. *Soc Net.* 2010; 32: 125–37.
4. Brusco M. An exact algorithm for a core/periphery bipartitioning problem. *Soc Net.* 2011; 33(1): 12–19.
5. Brusco M. Analysis of two-mode networks using nonnegative matrix factorization. *Soc Net.* 2011; 33(3): 201–210.
6. Everett MG, Borgatti SP. Peripheries of cohesive subsets. *Soc Net.* 1999; 21(4), 397–407.
7. Everett MG, Borgatti SP. Extending centrality. In Carrington PJ, Scott J, Wasserman S (Eds.), *Models and methods in social network analysis* (pp. 57–76). New York: Cambridge University Press, 2005.
8. Garcia Muniz AS, Ramos Carvajal C. Core/periphery structure models: an alternative proposal. *Soc Net.* 2006; 28(4): 442–48.
9. Csrmely P, London A, Wu L.-Y, Uzzi B. Structure and dynamics of core/periphery networks. *J Complex Networks* 2013; 1: 93–123.
10. Zhang X, Martin T, Newman MEJ. Identification of core/periphery structure in networks. *Phys Rev E* 2015; 91: 032803.
11. Borgatti SP, Everett MG, Freeman LC. *Ucinet for Windows, Version 6.153: Software for social network analysis.* Harvard, MA: Analytic Technologies, 2002.
12. Batagelj V, Ferligoj A, Doreian P. Direct and indirect methods for structural equivalence. *Soc Net.* 1992; 14: 63–90.
13. Doreian P, Batagelj V, Ferligoj A. *Generalized blockmodeling.* Cambridge, UK: Cambridge University Press, 2005.
14. Goldberg DE. *Genetic algorithms.* New York: Addison-Wesley, 1989.
15. Aarts E, Korst J, Michiels W. Simulated annealing. In Burke EK, Kendall G (Eds.) *Search methodologies* (pp. 187–210), New York: Springer, 2005.
16. Kernighan BW, Lin S. An efficient heuristic procedure for partitioning graphs. *Bell Sys Tech J.* 1970; 49: 221–6.
17. Van der Ark LA, Croon MA, Sijtsma K. Mokken scale analysis for dichotomous items using marginal models. *Psychometrika* 2008; 73(2): 183–208. <https://doi.org/10.1007/s11336-007-9034-z> PMID: 20046851
18. Wasserman S, Faust K. *Social network analysis: methods and applications.* Cambridge: Cambridge University Press, 1994.
19. Baker DR. A structural analysis of the social work journal network: 1985–1986. *J Soc Serv Res.* 1992; 15: 153–68.
20. Glover F, Laguna M. Tabu search. In Reeves C. (Ed.) *Modern heuristic techniques for combinatorial problems.* (pp. 70–141). Oxford: Blackwell, 1993.
21. Mladenović N, Hansen P. Variable neighborhood search. *Comp Oper Res.* 1997; 24: 1097–1100.
22. Cerny V. A thermodynamical approach to the traveling salesman problem. *J Optim Theory Appl.* 1985; 45: 41–51.
23. Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. *Science* 1983; 220: 671–80. <https://doi.org/10.1126/science.220.4598.671> PMID: 17813860
24. Metropolis N, Rosenbluth A, Rosenbluth M, Teller A, Teller E. Equation of state calculations by fast computing machines. *J Chem Phys.* 1953; 21: 1087–92.
25. Brusco MJ. A comparison of simulated annealing algorithms for variable selection in principal component analysis and discriminant analysis. *Comp Stat Data Anal.* 2014; 77(1): 38–53.
26. Lee JS, Park CH, Ebrahimi T. Theory and applications of hybrid simulated annealing. In Zelinka I, Snasel V, Abraham A (Eds.) *Handbook of optimization: from classical to modern approach* (pp. 395–422), Berlin: Springer-Verlag, 2012.
27. Nourani Y, Andresen B. A comparison of simulated annealing cooling strategies. *J Phys A: Math Gen.* 1998; 31: 8373–85.
28. Drabek TE, Tamminga HL, Kilijanek TS, Adams CR. *Managing multi-organizational emergency responses.* Boulder: University of Colorado, Institute of Behavioral Science, 1981.
29. Groenen PJF, Heiser WJ. The tunneling method for global optimization in multidimensional scaling. *Psychometrika* 1996; 61: 529–50.
30. Brewer DD. Forgetting in the recall based elicitation of personal and social networks. *Soc Net.* 2000; 22: 29–43.
31. Jeong H, Mason S, Barabási A-L, Oltvai N. Centrality and lethality of protein networks. *Nature* 2001; 411: 41–42. <https://doi.org/10.1038/35075138> PMID: 11333967