Neural, Parallel, and Scientific Computations 20 (2012) 327-348

CONTROL THEORY METHODOLOGY FOR SOCIAL NETWORKS: PREDICTION OF MISSING LINKS

N. G. MEDHIN AND G. L. PORTER

Department of Mathematics, North Carolina State University, Raleigh, NC

ABSTRACT. In this paper, we present a novel method for predicting missing links in a social network. The evolution of the social network is governed by a multiobjective optimal control problem (MOCP) where the dynamics is formulated based on social forces theory. The solution of the control problem is used to obtain an observed network structure as well as the initial conditions, parameters and constraints that led to its formation. From this observed network, we label some links as *known* and then identify a random subset of network connections and consider them as *unknown* or *missing*. Afterwards, a new MOCP is formed using the same network dynamics as before but now with constraints on the state to preserve the known links. In solving the new MOCP, we are able to reproduce existing links as well as predict or uncover the unknown or missing links. There are some advantages to this approach over those in the literature that rely heavily on network topology for link prediction. Within the MOCP framework, nodal attributes and past history is considered for link formation. This approach works for any given network structure and is capable of uncovering the qualitative, not just topological, reasons underlying link predictions unlike some other methods in the literature.

KEY WORDS: Multiobjective optimal control, missing links, Differential Evolution, social networks

1. Introduction

In this paper, we consider how to predict missing links within a social network. Missing links are those that are unobserved but are actually present within the social network [1]. The links may be unobserved for various reasons: it may be that the network itself is simply incomplete or it may be that the actors themselves attempt to hide their ties on purpose. Whatever the reason, predicting these unobserved links has garnered much attention from researchers in today's society. In particular, in the aftermath of the events of September 11, 2001, the capability to uncover missing links in terrorist and other criminal networks is believed to be crucial to national security. The ability to uncover hidden network relations provides a total picture of current and future interactions between entities within a social group.

In the literature, there are various methods used to predict missing networks links [5], [7], [10], [12], [2]. For instance, Liben-Nowell and Kleinberg (2004), [5], focus on

Received August 9, 2012

network topology such as common neighbors, degree, and shortest paths to predict links for actors within collaborations. Clauset, Moore, and Newman (2008), [1], use hierarchical structure of networks for link prediction. In their approach, Markov chains are used to sample hierarchical random graphs to fit the observed network. Pairs that have high average connection probability but appear unconnected in the observed network are identified as missing links. Our method for link prediction also starts with an observed network link structure but differs from those in the literature by not relying so heavily on network topology to predict missing links. Instead, our model uses constrained multiobjective optimal control and social force theory to predict missing links.

2. **Social Networks**

Several key concepts [13] form the basis of social network analysis and are fundamental to our study of social networks.

2.1. **Methodology for Social Networks.** *Nodes* form the basis of social networks and are often referred to as actors, actors or points depending on the context of discussion. Nodes in a social network can be social entities such as people, businesses, organizations, cities, nations, etc. An *edge* is a line connecting nodes. Edges are also referred to as links, ties, lines or arcs, representing a relationship or connection between a pair of nodes. In network analysis, there are many types of ties to include behavioral interaction ties (i.e., conversing or emailing), physical movement ties (i.e., migration) and individual evaluation ties (i.e., friendship among actors which is the focus of this paper). Network ties are often made based on some type of individual or entity attributes. *Attributes* describe characteristics of actors in a group. For example, for a friendship network, such attribute variables might include income potential, gender, race, sex, education level, political tendency, religious affiliation, marital status, etc. In fact, measurements on actors' attributes often constitute the make-up of social data and social networks.

There are two tools in particular which are often seen in the literature to represent social networks: *matrices* and *graphs*. In this work, we'll use both in illustrative examples of friendship networks. A *sociomatrix* is the primary matrix used in social network analysis and is denoted by $\mathbf{X}$. If there are N actors in a social group, then the sociomatrix for the group would be an $N \times N$ matrix of binary entries representing the relations between the actors. Each actors in the sociomatrix has a row and column both indexed $1, 2, \ldots, N$. The entries in the sociomatrix, x_{ij}, represent which nodes are linked. For our friendship model, relations in the sociomatrix may be directional and nondirectional which will lead to both symmetric and nonsymmetric sociomatrices. For symmetric sociomatrices, if two actors are friends, there will be a

1 in the ij-th and ji-th cells and a 0 if they're not friends. The ii-th cells will contain a value of 0 since actors do not befriend themselves. For nonsymmetric sociomatrices, while the ij-th cells may contain a 1, this may not be the case for the ji-th cell if the relation is not reciprocated.

A *graph* (often referred to as *digraph*) has a set of nodes representing the actors in the network and a set of lines to represent the existence of ties or links between pairs of actors. The graph can be drawn directly from the sociomatrix. Since relations in our model may or may not be symmetric, lines are both directional and nondirectional. In essence, if a directional line exists from actor i to j, it may not exist from j to i. We exclude any loops, which are lines between actors and themselves since actors do not befriend themselves.

2.2. **Social Forces Model for Social Networks.** Different modeling approaches have been developed to model social networks and social interaction. In this work, we take a more physical approach inspired by Helbing's social forces model for pedestrian walking behavior. We adapt Helbing's model to describe social interaction and ultimately, formulate a friendship model mathematically using the notion of social forces. In essence, actors interact as though they were subject to acceleration and repulsive forces when making their friendship choices. This approach assumes that individuals behave according to a set of rules in a manner that promotes their utility minimization, i.e, they choose courses of action with the most benefit and least cost. In the context of friendship networks, social forces theory assumes that each actor possesses a specific attitude toward making friends, a desire to befriend those who share their preferences and attributes and that they respect the private space of others. Consequently, following Helbing and Molnar's theory, these rules describing social interaction can be placed into a set of equations of motion [3].

2.2.1. *Assumptions.* We start with a fixed set of actors, denoted Λ, consisting of N actors, who begin as mutual strangers and enter into social relationships with other actors as time evolves. We make the following assumptions [4] in our model of network dynamics:

- All actors consider the same attributes when attempting to make friends.
- Actors do not change categories within a particular attribute.
- Relationships between actors depend on shared preferences for attributes and categories.
- Reciprocity for numerical preference levels is automatic by virtue of using the Euclidean distance as a measurement of closeness but this is not so for categorical preferences.
- Each actor attempts to maximize his status in the social group, i.e, he wishes to form as many relationships as possible.

- Finally, the objective functional of each actor decreases with an increase in shared attribute preferences and categories.

2.2.2. *Data.* The following data is required to run our model of network dynamics:
Data:

N – total number of actors in a social environment

m – total number of attributes under consideration

l – total number of categorical attributes under consideration

k – number of categories in a particular categorical attribute

$\mathbf{r}_i(t)$ – position vector describing actor i's preference for each attribute, $1, \ldots, m$

$\mathbf{y}_i$ – vector identifying various attribute categories to which actor i belongs

$\mathbf{w}_i$ – vector containing actor i's preferences for similar attribute categories

$\mathbf{v}_i^0$ – vector describing actor i's initial rate of change of attribute preferences at time $t = 0$

$\mathbf{v}_i(t)$ – vector describing actor i's rate of change of attribute preferences at time t

$\mathbf{u}_i(t)$ – vector describing actor i's control for each attribute, $1, \ldots, m$

Parameters:

l_{ij} – constant value set to ensure that actor j respects the private space of actor i

τ_i – relaxation time of actor i (a measure of how fast he returns to his $\mathbf{v}_i^0$)

$\mathcal{N}_i$ – reflects an actor's desire to stick to his belief system

Now that we have formally stated what each data variable represents, we can describe a few variables in more detail. For instance, $\mathbf{v}_i^0$ is meant to reflect how quickly a person intends to change their preference on a certain attribute in order to make friends; it is represented by a "velocity" vector in the social forces model described in Section 3.3 and hereafter, we will call it intended *social velocity*. Therefore, if a person intends to change their attribute preference levels rapidly, we'd expect to see a larger $\mathbf{v}_i^0$ compared to those who intend to change less rapidly. Similarly, $\mathbf{u}_i(t)$ controls how much actors vary their attribute preferences within a given set of bounds in order to make friends. The control variables of people who desire to make many friends will fluctuate greatly when compared to those actors who desire fewer relationships, reflected by control variables which are greatly restricted. Similarly, since l_{ij} controls how close actors allow others to get to them, those actors who desire to make many friends will have a larger value for l_{ij} than those who desire to keep others at a distance. Further, a large $\mathcal{N}_i$ is meant to penalize an actor for deviating from his belief system and thus results in an increase in an actor's performance index. Finally,

τ_i will be small for those who are more reluctant to change their attribute preferences permanently.

3. **The Model**

Our model for predicting missing links employs the same components as the social forces model for network dynamics with memory presented in an earlier paper [6] but adds additional constraints on the state.

3.0.3. *Model Components.* The model components for predicting missing links are as follows. First, we modify the performance index from its original version in [6] by adding penalties to remove the inequality constraints, $h_i(\mathbf{u})$:

$$\min_{\mathbf{u}} [J_1, \ldots, J_N] \tag{1a}$$

$$J_i = \Big[\sum_{j\neq i} \|\mathbf{r}_i(t_f) - \mathbf{r}_j(t_f)\|^2 + \sum_{j\neq i} \big\|\mathbf{w}_i(t_f)\cdot(\mathbf{y}_i(t_f) - \mathbf{y}_j(t_f))\big\|^2 + \mathcal{N}_i \int_{t_0}^{t_f} \|\mathbf{u}_i(t)\|^2\, dt\Big] \prod_{q=1}^{p} \xi_q^{c_q} \tag{1b}$$

where

$$\xi_q = \begin{cases} 1 + h_q(\mathbf{u}), & \text{if } h_q(\mathbf{u}) > 0, q = 1, \ldots, p \\ 1, & \text{otherwise} \end{cases}$$

and the constant $c_q = 10$.

The network dynamics remain unchanged:

$$\dot{\mathbf{r}}_i = \mathbf{v}_i \tag{1c}$$

$$\dot{\mathbf{v}}_i = \frac{1}{\tau_i}(\mathbf{v}_i^0 - \mathbf{v}_i) - \nabla_{\mathbf{r}_i} V_{int} - \nabla_{\mathbf{r}_i} V_m \tag{1d}$$

where

$$\begin{aligned} V_{int} = & \sum_{j\neq i} \|\mathbf{u}_i - \mathbf{u}_j\|^2 \\ & \cdot(1 + ((\|\mathbf{r}_i - \mathbf{r}_j\| + \|\mathbf{r}_i - \mathbf{r}_j - \mathbf{v}_j\Delta t\|)^2 - \|\mathbf{v}_j\Delta t\|^2)) \\ & \cdot \exp\{-l_{ij}((\|\mathbf{r}_i - \mathbf{r}_j\| + \|\mathbf{r}_i - \mathbf{r}_j - \mathbf{v}_j\Delta t\|)^2 - \|\mathbf{v}_j\Delta t\|^2)\} \end{aligned}$$

and

$$V_{mem}(\mathbf{r}_i, t) = \int_0^t \sum_{j\neq i} G(\mathbf{r}, s) \exp\Big\{\frac{t-s}{T}\Big\} ds$$

where

$$\begin{aligned} & G(\mathbf{r}, s) = -\gamma(1 + ((\|\mathbf{r}_i(t) - \mathbf{r}_j(s)\| + \|\mathbf{r}_i(t) - \mathbf{r}_j(s) - \mathbf{v}_j(s)\Delta t\|)^2 - \|\mathbf{v}_j(s)\Delta t\|^2) \\ & \cdot \exp\Big\{ - l_{ij}((\|\mathbf{r}_i(t) - \mathbf{r}_j(s)\| + \|\mathbf{r}_i(t) - \mathbf{r}_j(s) - \mathbf{v}_j(s)\Delta t\|)^2 - \|\mathbf{v}_j(s)\Delta t\|^2)\Big\} \end{aligned}$$

The same state and control bounds must be satisfied:

$$(\mathbf{r}_i(0) - \boldsymbol{\delta}_{i_{min}}) \leq \mathbf{r}_i(t) \leq (\mathbf{r}_i(0) + \boldsymbol{\delta}_{i_{max}}) \tag{1e}$$

$$-\boldsymbol{\delta}_{i_{min}} \leq \mathbf{u}_i \leq \boldsymbol{\delta}_{i_{max}} \tag{1f}$$

The following state constraints are added to the model in order to reproduce existing relations between actors which are known in advance:

$$d_{ij} \leq .8d_{avg}, \quad \text{if } i, j \text{ are linked} \tag{1g}$$

$$d_{ij} > .8d_{avg}, \quad \text{if } i, j \text{ are not linked} \tag{1h}$$

The social distance between actors, d_{ij}, and the average distance, d_{avg}, are calculated as follows:

$$d_{ij} = \sum_{j \neq i} \|\mathbf{r}_i(t) - \mathbf{r}_j(t)\|^2 + \sum_{j \neq i} \left\|\mathbf{w}_i(t) \cdot (\mathbf{y}_i(t) - \mathbf{y}_j(t))\right\|^2$$

$$d_{avg} = \frac{\sum_i \sum_{j \neq i} d_{ij}}{N^2 - N}$$

In the missing link model, the most significant difference from the original model in [6] are the added state constraints and the modified performance index to deal with these new constraints. The constraints exist on the state variables to ensure that we reproduce known relations as well as uncover the missing links. In essence, we want there to be links between those people who were already friends and we don't want links between those who were not friends. By using known information on parameters and data for existing links, the new model uncovers missing link information.

4. **Pareto Optimality**

Most likely the objective functions in the above MOCP are competing objectives which will make it difficult to minimize them all at once; yet, if it happens that a single solution is found for the MOCP, then the objectives are really not competing after all. That said, since no single minimum is likely to be found, the concept of *optimality* for multiobjective optimal control problems with vector-valued cost must be defined. Once again, our definition of optimality in the multiobjective framework is **Pareto optimality**.

A solution $\mathbf{u}^*$ dominates $\mathbf{u}$ if and only if $J_i(\mathbf{u}^*) \leq J_i(\mathbf{u})\ \forall\ i \in \{1, 2, \ldots, s\}$ and $J_i(\mathbf{u}^*) < J_i(\mathbf{u})$ for at least one $i \in \{1, 2, \ldots, s\}$. The set of nondominated points from the search space form Pareto front or Pareto optimal set.

Definition 4.1. For a given vector of objective or cost functions $\mathbf{J}(u) = [J_1(u), J_2(u), \ldots, J_s(u)]$, the control u^* is **Pareto optimal** if there does not exist $\mathbf{u}$ such that

$$J_i(\mathbf{u}) \leq J_i(\mathbf{u}^*)$$

and for at least one i, $i \in \{1, 2, \ldots, s\}$, we get

$$J_i(\mathbf{u}) < J_i(\mathbf{u}^*)$$

Evolutionary algorithms (EA), like Differential Evolution, are well-suited for solving multiobjective optimization problems since they are capable of providing a Pareto optimal set in a single run.

5. Numerical Methods

5.1. **Differential Evolution.** Differential Evolution (DE) is a population-based search method developed by Storn and Price [9] to handle problems with multiple objectives over continuous domains. DE is an appealing approach for solving MOCPs because it eliminates the need to consider function continuity, convexity, or concavity unlike some traditional search techniques where the complexities must be given great attention. In addition, DE is capable of providing a complete set of Pareto-optimal solutions in a single run [8]. It is a stochastic population-based direct search method that improves some randomly generated initial population through *mutation, crossover*, and *selection*. The algorithm includes the following steps.

5.1.1. *Steps for Differential Evolution (DE) Algorithm.*

- **Step 1:** Random Population Initialization
  In this step, $\mathbf{u}^g_{j,i}$ means the i-th entry of the vector $\mathbf{u}^g_j$. We initialize the population as follows:

  $$\mathbf{u}^g_{j,i} = \mathbf{u}^g_{j,i_{min}} + rand() * (\mathbf{u}^g_{j,i_{max}} - \mathbf{u}^g_{j,i_{min}}), \quad j = 1, 2, \ldots, NP,$$

  g is the current generation and $rand()$ is a random number in $[0, 1)$. The i-th component of the vector $\mathbf{u}^g_j$, $j = 1, 2, \ldots, NP$, has a lower bound, $\mathbf{u}^g_{j,i_{min}}$, and an upper bound, $\mathbf{u}^g_{j,i_{max}}$.
- **Step 2:** Mutation
  For each $j = 1, 2, \ldots, NP$, pick $j_1, j_2, j_3 \in \{1, 2, ..., NP\}$ randomly and form the vector $\hat{\mathbf{z}}^g_j$ according to the formula:

  $$\hat{\mathbf{z}}^g_j = \mathbf{u}^g_{j_1} + W * (\mathbf{u}^g_{j_2} - \mathbf{u}^g_{j_3}), \quad j = 1, 2, \ldots, NP$$

  where j_1, j_2, j_3 are mutually different and not equal to j. The parameter W is a scaling factor for mutation and is usually a value between 0 and 1.

- **Step 3:** Crossover
  As in Step 1, we denote the i-th component of the vector z_j^g by $z_{j,i}^g$. The operation *crossover* is implemented as follows:

$$z_{j,i}^g = \begin{cases} u_{j_1,i}^g + W * (u_{j_2,i}^g - u_{j_3,i}^g) & \text{if } rand() < CR \text{ or } i = \hat{i}, \\ u_{j,i}^g & \text{otherwise} \end{cases}$$

  where $\hat{i}$ is a randomly selected index from $\{1, 2, \ldots, D\}$.
- **Step 4:** Selection

$$\mathbf{u}_j^{g+1} = \begin{cases} \mathbf{z}_j^g & \text{if } J(\mathbf{z}_j^g) \leq J(\mathbf{u}_j^g), \\ \mathbf{u}_j^g & \text{otherwise} \end{cases}$$

- **Step 5:** Termination criteria in the literature often includes running the algorithm for some maximum number of generations or until some desired objective function value is reached.

5.2. **Parallel Differential Evolution.** There are several variations of Parallel Differential Evolution [11] found in the literature and here we have modified and merged the different ones into one suitable for our problem. We implement our version as follows:

- **Step 1:** Request K nodes (or processors) taking one node to be the master node.
- **Step 2:** At the master node, create K-1 populations and send one to each of the remaining K-1 nodes.
- **Step 3:** At each of the K-1 nodes, each population evolves toward a nondominated set using DE.
- **Step 4:** As the termination criteria is met, each node sends its nondominated set to the master node.
- **Step 5:** At the master node, compare the K-1 nondominated sets to get the final Pareto-optimal set.

6. Computer Simulation

6.1. **Problem Formation.** To demonstrate the capabilities of the missing link model, we solve the MOCP in (1a)–(1h) with $N = 25$ actors and five attributes. We use the sociomatrix in Figure 1 as our observed network. From this matrix, we discover two disjoint cliques: Clique 1: {5,6,12} and Clique 2: {9,11,16,23,23,25}. Relations between members in these cliques will serves as the *known* links. We create an additional set of actors chosen randomly from the sociomatrix and denoted them by M: {2,4,7,10,14}. We pretend the internal links between actors in M as well external links between actors in M and actors in the cliques are *unknown*. The objective now

is to try to reproduce known relations amongst members in the cliques while simultaneously uncovering the relations between the cliques and actors in the set M using the information we already have on existing links and parameters.

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

FIGURE 1. Sociomatrix with Interaction Potential, V_{int} (N = 25)

6.2. **Implementation.** To start, we must assume that we have some observable data on the nodes in set M for whom we do not know the links and wish to identify them. Therefore, we assume we know their initial attribute data to include preferences and categories. From this information, we infer their similarity weights, $\mathbf{w}_i$, and intended motivation toward making friends, $\mathbf{v}_i^0$.

For actors within cliques 1 and 2, we use their known parameter values but we estimate the parameter values of those actors in set M by permitting them to fall within the allowable limits previously established. For instance, in determining the previous sociomatrix, certain maximum and minimum bounds were identified for parameters, l_{ij} and τ_i:

$$0.05 \leq l_{ij} \leq 0.25$$

$$5.0 \leq \frac{1}{\tau_i} \leq 15.0.$$

There are several state constraints which must be satisfied in order to maintain the existing links between actors in Clique 1. Here is an example of such constraints

using the smaller clique:

$$d_{5,6} \leq .8d_{avg}$$
$$d_{5,12} \leq .8d_{avg}$$
$$d_{6,5} \leq .8d_{avg}$$
$$d_{6,12} \leq .8d_{avg}$$
$$d_{12,5} \leq .8d_{avg}$$
$$d_{12,6} \leq .8d_{avg}$$

Similar constraints must be satisfied to maintain the existing links between actors belonging to Clique 2 as well.

In addition, the model must ensure that there are no links between members belonging to the different cliques. For example, here are the constraints that must be met for actor 5 from Clique 1 to maintain his "no link" status to actors in Clique 2:

$$d_{5,9} > .8d_{avg}$$
$$d_{5,11} > .8d_{avg}$$
$$d_{5,16} > .8d_{avg}$$
$$d_{5,22} > .8d_{avg}$$
$$d_{5,23} > .8d_{avg}$$
$$d_{5,25} > .8d_{avg}$$

Constraints similar to these must be established and satisfied for each actor belonging to the two cliques.

Once the existing links as shown in Table 1 have been established in the manner suggested, attention can then be focused on filling in the missing links between actors in the cliques and actors in set M. To do so, we again construct the multiobjective optimal control problem (MOCP) in similar fashion as before but subject to the additional state constraints used to ensure the existing relationships. Finally, a Pareto optimal solution of this newly formed MOCP as outlined in equations 1 is used to construct the sociomatrix and identify the existing links as well as the missing links.

6.3. **Numerical Results and Analysis.** To solve the MOCP, we used Parallel DE as outlined above with slight modifications to the basic DE algorithm. For those actors in set M, the parameters, l_{ij} and τ_i, are treated as control variables and allowed to evolve as a population using the random search method. Specifically, in the basic Differential Evolution scheme, the initialization and mutation steps were modified for parameters, l_{ij} and τ_i. For the sake of clarity in the following modifications, we

drop the j from the l_{ij} parameter and just use l_i; this notation does not change the parameter's definition.

In the **initialization** step, we added the following equations:

$$l^g_{k,i} = l^g_{k,i_{min}} + rand() * (l^g_{k,i_{max}} - l^g_{k,i_{min}})$$

and

$$\tau^g_{k,i} = \tau^g_{k,i_{min}} + rand() * (\tau^g_{k,i_{max}} - \tau^g_{k,i_{min}})$$

where $rand()$ is a uniformly distributed random number $\in [0, 1)$ and $l^g_{k,i_{min}}$ and $l^g_{k,i_{max}}$ are lower and upper bounds respectively on the i-th component of the k-th vector, $k = 1, 2, \ldots, NP$.

In the **mutation** step, in addition to the existing control vectors, for each of the population vectors, $\boldsymbol{l}_k$ and $\boldsymbol{\tau}_k$, $k = 1, \ldots, NP$, Differential Evolution would generate competing trial vectors, $\hat{\boldsymbol{l}}_k$ and $\hat{\boldsymbol{\tau}}_k$:

$$\hat{\boldsymbol{l}}^g_k = \boldsymbol{l}^g_{j_1} + W * (\boldsymbol{l}^g_{j_2} - \boldsymbol{l}^g_{j_3})$$

and

$$\hat{\boldsymbol{\tau}}^g_k = \boldsymbol{\tau}^g_{j_1} + W * (\boldsymbol{\tau}^g_{j_2} - \boldsymbol{\tau}^g_{j_3,i})$$

where j_1, j_2, and j_3 are random mutually different vectors belonging to $[0, NP]$ and not equal to vector k.

To solve the problem, we used Parallel DE as outlined in subsection 5.2 with the following criteria:

1. **Requested number of nodes**: 61
2. **DE parameters:** $NP = 30$ per node, $W = 0.5$, and $CR = 0.5$
3. **Termination Criteria:**

$$\sum_{i=1}^{25} \left| \frac{J_i^{(k)}(\mathbf{u}^{(1)}) + \cdots + J_i^{(k)}(\mathbf{u}^{(NP)})}{NP} - \frac{J_i^{(k-1)}(\mathbf{u}^{(1)}) + \cdots + J_i^{(k-1)}(\mathbf{u}^{(NP)})}{NP} \right| < 10^{-5}$$

We used the fourth order Runge Kutta method to integrate the state equations. We handled the bounds on the state and control vectors by choosing the controls appropriately to satisfy both. In order to avoid problems in distinguishing the added state constraints for link prediction, we had to make modifications in order to solve the problem:

$$d_{ij} + \epsilon \leq .8 d_{avg}, \text{ if } i, j \text{ are linked,} \tag{2}$$

and

$$d_{ij} + \epsilon > .8 d_{avg}, \text{ if } i, j \text{ are not linked.} \tag{3}$$

Essentially, we simply adjust these constraints by some small amount, $\epsilon = .1*\min(d_{ij})$, in order to solve the problem. We believe this slight adjustment does not impact the accuracy of link prediction. Parallel DE was implemented in C++ and took approximately ? hours to generate a set of Pareto optimal points.

In Table 2, the sociomatrix for members in the cliques and set M is shown. The model was able to reproduce the known relations as well as fill in the blanks for the unknown relations. When compared to the observed network in the sociomatrix from Figure 1, which shows the actual links between actors, our link prediction model is 100% accurate in predicting missing links. Figure 2 shows the associated digraph for the predicted links.

TABLE 1. Sociomatrix with Missing Links

Actor	2	4	5	6	7	9	10	11	12	14	16	22	23	25
2	0													
4		0												
5			0	1		0		0	1		0	0	0	0
6			1	0		0		0	1		0	0	0	0
7					0									
9			0	0		0		1	0		1	1	1	1
10							0							
11			0	0		1		0	0		1	1	1	1
12			1	1		0		0	0		0	0	0	0
14										0				
16			0	0		1		1	0		0	1	1	1
22			0	0		1		1	0		1	0	1	1
23			0	0		1		1	0		1	1	0	1
25			0	0		1		1	0		1	1	1	0

7. **Clique Expansion and Infiltration**

7.0.1. *Introduction.* In this section, we leverage what was learned concerning clique formation in [6] to explore clique infiltration. Previously, it was discovered that clique formation requires mutual affection amongst actors which is based on shared attribute preferences and categories as well as similar choices for the various model parameters. To build on that knowledge, the goal of this section is to try to determine under what circumstances existing cliques would allow other actors to join them. Alternatively, the goal can be restated as how to forcibly insert certain actors into cliques.

When looking across the row labeled 10 in Table 2, we see clearly that actor 10 has directional relations or perceived closeness on his part toward members of

TABLE 2. Sociomatrix with Predicted Links

Actor	2	4	5	6	7	9	10	11	12	14	16	22	23	25
2	0	0	0	0	0	0	0	0	1	0	1	1	1	1
4	0	0	1	1	1	0	1	0	1	0	0	0	0	0
5	0	0	0	1	0	0	0	0	1	0	0	0	0	0
6	0	0	1	0	0	0	0	0	1	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	1	1	0	1	1	1	1	1
10	0	0	1	1	0	0	0	1	1	0	0	0	0	0
11	0	0	0	0	0	1	1	0	0	1	1	1	1	1
12	0	0	1	1	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	1	0	1	0	0	0	1	0	1
16	1	0	0	0	1	1	1	1	0	1	0	1	1	1
22	0	0	0	0	1	1	0	1	0	1	1	0	1	1
23	1	0	0	0	0	1	0	1	0	1	1	1	0	1
25	0	0	0	0	1	1	0	1	0	1	1	1	1	0

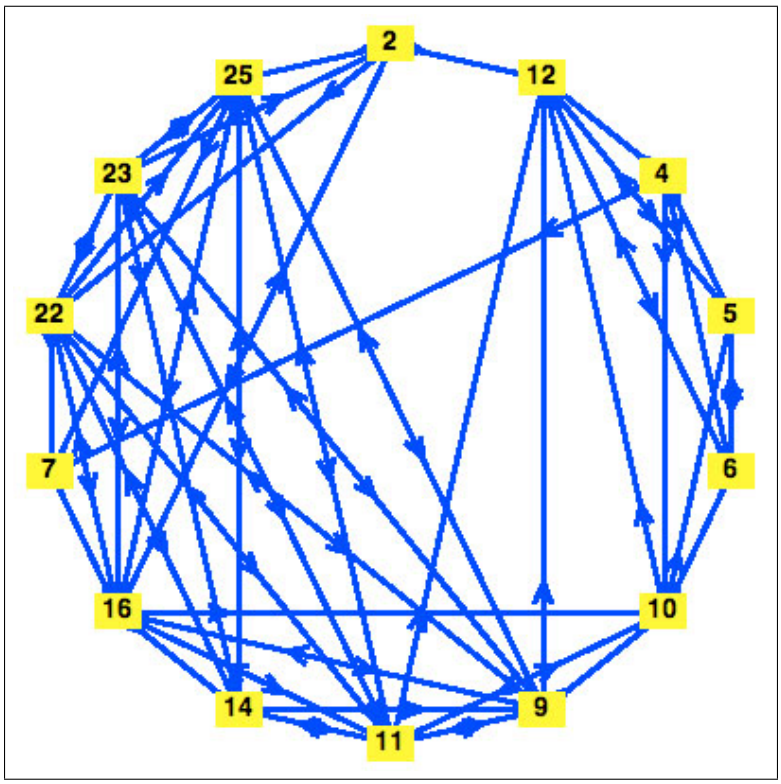


FIGURE 2. Digraph for Predicted Links

Clique 1. Yet, this affection from actor 10 is not shared by Clique 1 members as shown by looking down the column labeled 10 in the table. This lack of reciprocity

is further confirmed by the social distance plotted in Figure 3(a). It is interesting to see what choice of parameters on the part of actor 10 will allow Clique 1 members to show reciprocity thus allowing actor 10 to infiltrate the clique.

7.1. **Infiltration of Clique 1.** To start the process, we implement a slight change to the model by deleting the *memory* potential, V_{mem}, in equation (1d). This should allow individuals more freedom to interact since actors in Clique 1 will base their friendship decision on current information instead of actor 10's entire history.

Suppose that in addition to the constraints in (1), we add the below constraints to the model in an attempt to try to force Clique 1 to accept actor 10. That is, we use

$$\begin{aligned} d_{10,5} &\leq .8d_{avg} \\ d_{10,6} &\leq .8d_{avg} \\ d_{10,12} &\leq .8d_{avg} \\ d_{5,10} &\leq .8d_{avg} \\ d_{6,10} &\leq .8d_{avg} \\ d_{12,10} &\leq .8d_{avg} \end{aligned}$$

Afterwards, we initialize actor 10's parameters, l_{ij} and τ_i, to values ranging between the previously identified minimum and maximum values and allow them to evolve as a population along with the control parameters using Differential Evolution. In this manner, we hope to discover whether or not certain choices for the various parameters guarantee membership in a particular clique.

Once again, the same algorithmic criteria as before was used to solve the problem. The result was that actor 10 did not immediately become a member of Clique 1 given changes to his parameters values, l_{ij} and τ_i. After several more attempts and even modifying other parameters like actor 10's similarity weights, $\mathbf{w}_i$, and attitude toward making friends, $\mathbf{v}_i^0$, actor 10 was still unsuccessful in penetrating the clique. A thorough review of the raw data indicates that actor 10 actually has many things in common with members of the clique. In fact, actor 10 is in the same age group and shares the same political preference as members of the clique. While they all share similar education and income preferences as well as views on tolerating diversity, actor 10 belongs to a different religious category than clique members. Significantly, it turns out that Clique 1 is very strongly aligned when it comes to religious preference evident by the fact that all of its members have the highest similarity weight possible, 1.0, for this particular attribute preference.

Tables 3–7 show the social distance between Clique 2 and actor 10 by attribute and allow us to take a microscopic look at their preferential differences. Analyzing

these tables confirms what we learned from reviewing the raw data. Indeed, Clique 2 members are strongly aligned in all attribute preferences especially religious preference. We highlight the large difference in religious preference between Clique 2 members and actor 10 in Table 7.

Undoubtedly, we have discovered the primary reason for actor 10's failure, thus far, to successfully penetrate the clique. Since it is so important to members of Clique 1 to only associate with actors of their same religious category, actor 10 may have to take some drastic measures to successfully penetrate the clique. Suppose we relax one of the basic assumptions of the model as discussed in subsection **??** and allow actor 10 to change his religious category. With this change, members of the clique finally find actor 10 appealing enough to reciprocate his friendship. This mutual affection is captured in Figure 3 which graphs the social distance between Clique 1 and actor 10 before and after the change in category. Actor 10 successfully enters Clique 1 with evolved parameter values $l_{ij} = 0.177365$ and $\tau_i = 1/8$ which is within the range of values used by the rest of the members in Clique 1.

TABLE 3. Distance between Education Preferences for actors in Clique 1

Actor	j			
i	5	6	10	12
5	0	0.0014	0.8574	0.8638
6	0.0014	0	0.8560	0.8624
10	0.5074	0.5060	0	0.0064
12	0.5138	0.5124	0.0064	0

TABLE 4. Distance between Age Preferences for actors in Clique 1

Actor	j			
i	5	6	10	12
5	0	0.0062	0.0043	0.0090
6	0.0062	0	0.0020	0.0152
10	0.0043	0.0020	0	0.0132
12	0.0090	0.0152	0.0132	0

7.2. **Infiltration of Clique 2.** As for Clique 2, we repeat the experiment using actor 14 with much success and far greater ease than with Clique 1. The primary reason for this is that Clique 2 is very tolerant of others which is evident by their choice of parameters, in particular, their similarity weights. If we look down column labeled 14 in Table 2, it is clear that every member in Clique 2 has directional ties toward actor 14, which means they already consider him their friend. However, the

TABLE 5. Distance between Income Preferences for actors in Clique 1

Actor	j			
i	5	6	10	12
5	0	0.0025	0.0007	0.0216
6	0.2525	0	0.0018	0.2691
10	0.2507	0.0018	0	0.2709
12	1.0216	0.5191	0.5209	0

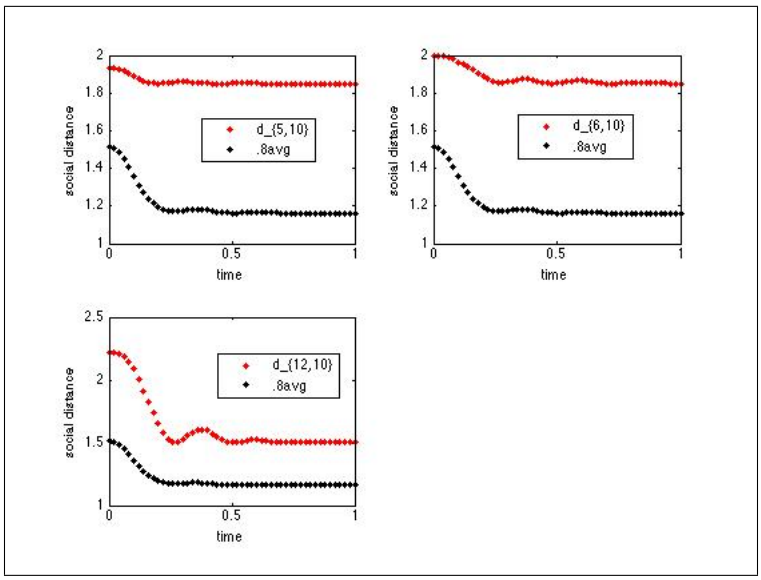
TABLE 6. Distance between Political Preferences for actors in Clique 1

Actor	j			
i	5	6	10	12
5	0	0.0004	0.0013	0.0258
6	0.0004	0	0.0008	0.0262
10	0.0013	0.0008	0	0.0271
12	0.0258	0.0262	0.0271	0

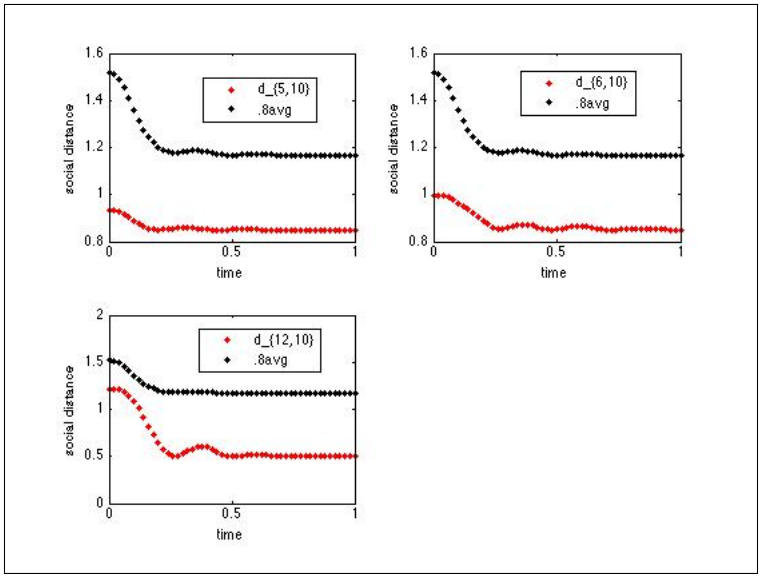
TABLE 7. Distance between Religious Preferences for actors in Clique 1

Actor	j			
i	5	6	10	12
5	0	0.0005	**2.0019**	0.0050
6	0.0005	0	**2.0013**	0.0045
10	0.5019	0.5013	0	0.5031
12	0.0050	0.0045	**2.0031**	0

problem in this experiment is that actor 14 is not friendly toward all members in the clique, in particular, actors 16 and 23, from looking across the row labeled 14 in the same table. While actor 14 shares numerous parameters, preferences, and categories with Clique 2 members, his beliefs on diversity initially prevent him from entering the clique as indicated by Figure 4(a). Once again, breaking out the social distance between Clique 2 members and actor 14 in Tables 8 and 12 allows us to take a microscopic look at their relations. When analyzing the tables , we focus on the distance between actors 14, 16, and 23. We discover that actor 14 may need to reduce his large similarity weights for age preference and political preference to reflect more tolerance for diversity. These changes allow him to easily infiltrate Clique 2 as supported by the before and after pictures in Figure 4. Actor 14 successfully enters Clique 2 with evolved parameter values $l_{ij} = 0.234454$ and $\tau_i = 1/12$ which is within the range of values used by the rest of the members in Clique 2.



(a) Before Infiltration



(b) After Infiltration

FIGURE 3. Social Distance between Clique 1 and Actor 10

8. **Conclusion**

In this paper, we discovered that multiobjective optimal control coupled with social forces theory provides a suitable framework for uncovering missing links within social networks with reasonable accuracy. We were successful in our approach to use known information regarding existing network links to predict hidden links. We also gained insight as it relates to clique infiltration. In fact, the clique expansion experiments indicate that the model is performing as designed which is very reassuring. At times, actors were able to relate to each other on shared attribute preferences alone; yet, at other times, shared preferences for attributes did not seem to be enough

TABLE 8. Distance between Education Preferences for actors in Clique 2

Actor	j						
i	9	11	14	16	22	23	25
9	0	0.0016	0.0008	0.5007	0.5026	0.5013	0.5062
11	0.0016	0	0.0008	0.5009	0.5010	0.5002	0.5078
14	0.0008	0.0008	0	0.5001	0.5018	0.5005	0.5070
16	0.2507	0.2509	0.2501	0	0.0019	0.0007	0.0069
22	0.2526	0.2510	0.2518	0.0019	0	0.0012	0.0088
23	0.2513	0.2502	0.2505	0.0007	0.0012	0	0.0075
25	0.2562	0.2578	0.2570	0.0069	0.0088	0.0075	0

TABLE 9. Distance between Age Preference actors in Clique 2

Actor	j						
i	9	11	14	16	22	23	25
9	0	0.0028	0.0062	0.0028	0.0055	0.0067	0.0054
11	0.0028	0	0.0034	0.0000	0.0027	0.0039	0.0082
14	0.0062	0.0034	0	0.0034	0.0007	0.0005	0.0116
16	0.0028	0.0000	0.0034	0	0.0027	0.0039	0.0082
22	0.0055	0.0027	0.0007	0.0027	0	0.0012	0.0109
23	0.0067	0.0039	0.0005	0.0039	0.0012	0	0.0120
25	0.0054	0.0082	0.0116	0.0082	0.0109	0.0120	0

TABLE 10. Distance between Income Preferences for actors in Clique 2

Actor	j						
i	9	11	14	16	22	23	25
9	0	0.0009	0.0013	0.0003	0.0006	0.0002	0.0055
11	0.0009	0	0.0004	0.0006	0.0002	0.0011	0.0047
14	0.0013	0.0004	0	0.0010	0.0007	0.0015	0.0042
16	0.0003	0.0006	0.0010	0	0.0003	0.0005	0.0052
22	0.0006	0.0002	0.0007	0.0003	0	0.0009	0.0049
23	1.0002	1.0011	1.0015	1.0005	1.0009	0	1.0058
25	0.0055	0.0047	0.0042	0.0052	0.0049	0.0058	0

to make connections. For instance, the numerous failures of actor 10 to penetrate Clique 1 reflect the impact of categorical differences on friendship choices. The model

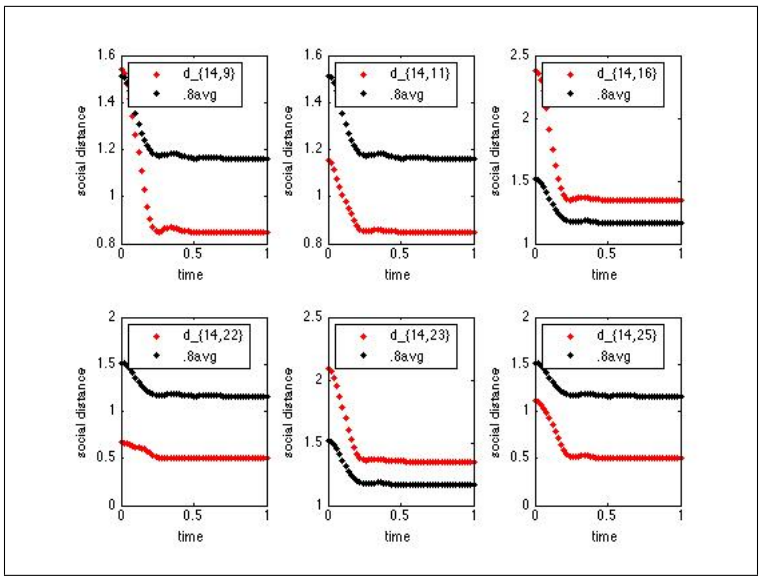
TABLE 11. Distance between Political Preferences for actors in Clique 2

Actor	j						
i	9	11	14	16	22	23	25
9	0	0.0012	0.0010	0.0009	0.0035	0.0028	0.0064
11	0.5012	0	1.0002	0.5003	1.0022	0.5016	1.0076
14	2.5510	1.7002	0	2.5501	0.0024	2.5517	0.0074
16	0.0009	0.0003	0.0001	0	0.0025	0.0018	0.0073
22	2.5535	1.7022	0.0024	2.5525	0	2.5507	0.0098
23	0.0028	0.0016	0.0017	0.0018	0.0007	0	0.0091
25	2.5564	1.7076	0.0074	2.5573	0.0098	2.5591	0

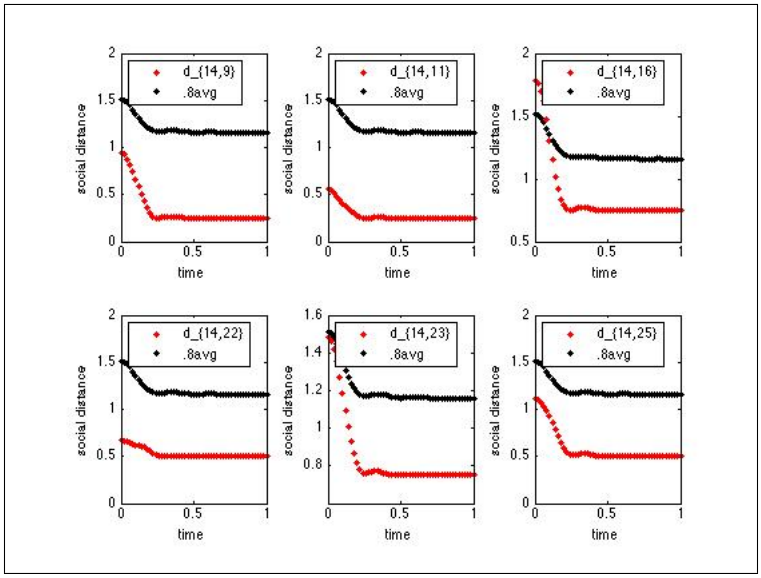
TABLE 12. Distance between Religious Preferences for actors in Clique 2

Actor	j						
i	9	11	14	16	22	23	25
9	0	0.0016	0.0000	0.0008	0.0013	0.0003	0.0025
11	0.0016	0	0.0017	0.0008	0.0003	0.0019	0.0041
14	0.0000	0.0017	0	0.0009	0.0014	0.0002	0.0025
16	0.0008	0.0008	0.0009	0	0.0005	0.0011	0.0033
22	0.0013	0.0003	0.0014	0.0005	0	0.0016	0.0039
22	0.0003	0.0019	0.0002	0.0011	0.0016	0	0.0022
25	0.0025	0.0041	0.0025	0.0033	0.0039	0.0022	0

highlights the significant role that attitudes toward diversity can play in making connections via its similarity measures which account for categorical preferences thus adding an element of realism.



(a) Before Infiltration



(b) After Infiltration

FIGURE 4. Social Distance between Actor 14 and Clique 2

REFERENCES

[1] A. Clauset, C. Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. report, University of Michigan, mejn@umich.edu, 2008.

[2] M. A. Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. report, Rensselaer Polytechnic Institute, falhasan, chaojv, salems, zakig@cs.rpi.edu, 2005.

[3] D. Helbing and P. Molnar. Social forces model for pedestrian dynamics. *Physics Review*, E(51):4282–4286, 1995.

[4] D. Helbing, P. Molnar, and F. Schweitzer. Computer simulation of pedestrian dynamics and trail formation. May 1998.

[5] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. report, Massachusetts Institute of Technology, dln@theory.lcs.mit.edu, 2004.

[6] N. G. Medhin and G. L. Porter. Constrained multiobjective control problems: Application to social networks. *(Submitted: Nonlinear Analysis Series A: Theory, Method & Applications)*, 2008.

[7] A. Popescul and L. H. Ungar. Statistical relational learning for link prediction. report, University of Pennsylvania, popescul,ungar@cis.upenn.edu, 2003.

[8] R. Sarker and H.A. Abbass. Differential evolution for solving multi-objective optimization problems. report, University Of New South Wales, Northcott Drive, Canberra ACT, 2600, Australia.

[9] R. Storn and K. V. Price. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. *Computer-Mediated Communication*, 11((2006)):1062–1084, 2006.

[10] B. Taskar, Ming-FaiWong, P. Abbeel, and D. Koller. Link prediction in relational data. report, Stanford University, btaskar, mingfai.wong, abbeel, koller @cs.stanford.edu, 2005.

[11] D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, and M. N. Vrahatis. Parallel differential evolution. Technical report, Greece.

[12] C. Wang, V. Satuluri, and S. Parthasarathy. Local probabilistic models for link prediction. Technical report, The Ohio State, 2007.

[13] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University, New York, NY, 1994.