A Hybrid Cuckoo Search for Direct Blockmodeling

Saeed NasehiMoghaddam*
Faculty of Engineering,University of Zanjan, Zanjan, Iran
nasehi@znu.ac.ir
Mehdi Ghazanfari
Faculty of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran
mehdi@iust.ac.ir
Babak Teimourpour
Faculty of Systems and Industrial Engineering, Tarbiat Modares University, Tehran, Iran
b.teimourpour@modares.ac.ir

Received: 18/Oct/2016 Revised: 20/May/2017 Accepted: 27/May/2017

Abstract

Block modeling as a social structure discovery process needs to find and adopt a partitioning of actors to equivalent classes or positions. The best partitioning, naturally, must provide the closest estimation of network ties and show the most agreement with original network data. This interpretation of the best, leads to the structure with the most fitness to original network data. Finding this best partition vector can be formulated as an optimization problem and can be solved by Meta heuristic algorithms. In this paper, we use cuckoo search and genetic algorithm as a basis for comparison with cuckoo search. In addition to simple cuckoo search, we apply a hybrid cuckoo search algorithm to find the solution. The results of experiments through multiple samples reveals that while genetic algorithm shows the better performance in terms of convergence time and small iteration, the hybrid cuckoo search finds the better solutions than genetic algorithm in large iteration in terms of quality of solutions measured by fitness function. Furthermore, the hybrid cuckoo search shows no significant superiority over the simple cuckoo search, unless the large iteration numbered is used. In addition to block model problem, the proposed hybrid cuckoo search shows clear superiority over the greedy discrete PSO for community detection problem .

Keywords: Social Network Analysis (SNA): blockmodeling: Genetic Algorithm: Cuckoo Search: likelihood ratio statistics G^2.

1. Introduction

The social structure of a given network is discovered during blockmodeling process in the framework of positional analysis. During this process using a partitioning vector of actors to equivalent classes, called positions, network relational data is summarized to reduced network of structural positions which is assumed as the structure of a given network. This structure can be shown by density/image matrices or reduced graphs and reflects the whole original network of actors in the simplified small network of structural positions. Thus structure discovery process, similar to community detection process, depends on the partition vector of actors to positions, but a community is not assumed as a structural position. As mentioned by[1], the term “community” was used by “physicists” instead of term “cohesive group” which “is foundational in sociology”. It is clear that grouping actors based on cohesion, does not necessarily means that members of a cohesive subgroup have similar ties to/ from the other actors. However if the structure of a given network discovered during blockmodeling is “cohesive subgroup” then the structural positions of that network will correspond to communities of the network. The partitioning of actors to structural positions is done so that all actors of a position are equivalent and similar in terms of their relational pattern. In conventional blockmodeling, several definitions of equivalency were used to find this partition vector. Pioneer of them is structural equivalence (SE), introduced by [2], which implies that two actors belong to the same structural equivalent class iff they have identical ties to / from all other actors. In addition to SE, automorphic, isomorphic and regular equivalence were proposed by researchers, and for each definition, an algorithm for partitioning actors to positions was proposed by researchers. Since for social structure discovery of a given network, the original network data is summarized by actor partitioning, and the inconsistent partition vectors produced by these equivalence definitions, leads to several structures for the same network and the question is that which of these structures is the best representative of the network?, and which equivalence definition is suitable for finding the structure of a network?, and a fundamental question is that is it necessary to use an equivalence definition for structure discovery purpose?, if not is it possible to adopt a computational approach in the form of an optimization problem to find the best social structure of a given network?, if so how to solve this optimization problem?

In this paper, we consider the social structure discovery as an optimization problem and intend to find the best social structure of a network. The best structure, naturally, provides the best estimation of network relational data. Since the social structure of a given network, depends on a partition vector of actors to

---

* Corresponding Author

positions, a partition vector, producing the closest estimation of network relational data, prepares the best social structure of a given network. This interpretation of the best, which is mentioned in none of the previous contributions of optimized block model problem, is the basis of our definition of block model problem. In order to solve our definition of block model problem, we employ two of Meta heuristics: cuckoo search and genetic algorithm as a basis for comparison with cuckoo search. In addition to simple cuckoo search, we apply a hybrid cuckoo search algorithm to find the solution. Now in the next section we briefly review the previous contributions pertinent to the optimal block model problem and then in section 3, we propose our definition for the block model problem along with algorithms used to optimize this problem and in section 4 the results of experiments for investigating the performance of the proposed algorithm are presented and finally the results summarized at section 5.

2. Related Works

In this section we review contributions relevant to applying optimization approach in block modeling and then a brief introductory subsection for meta heuristic algorithms generally and for cuckoo search algorithm specifically is provided.

2.1 Optimization Approaches In Block Modeling

While conventional block modeling allocates actors to positions by clustering, based on distances of actors from each other according to several equivalence definitions, such as structural and regular equivalence, generalized blockmodeling procedure of [3], using a local optimization procedure, searches the best partition vector conforming with a predetermined social structure. In fact, in generalized blockmodeling, actor partitioning is done without need to any equivalence definition, and the social structure of network is assumed known; therefore the structure of network is not sought. The independency from equivalence definitions provides direct approach for generalized blockmodeling. This generalization of blockmodeling could be adopted for two-mode network by [4], signed network by [5], sparse network by [6], multilevel network by [7], and valued networks by [8] , which were implemented in Pajek software and blockmodeling R package. However if the predefined social structure is unknown, the applicability and feasibility of generalized blockmodeling will be limited.

Besides generalized blockmodeling, the other direct approaches to structure discovery in the framework of an optimization problem have recently been proposed by authors. James, et al. [9] suggested a novel genetic algorithm with some ad hoc operators, for blockmodeling and showed its applicability in large network. But the usage of Herfindahl-Hirschman Index ([10]; [11]), HHI, which is just the sum of squared position sizes over all positions, as fitness function is not suitable in all cases such as valued relations; because maximized (HHI) prepares solution with high dense diagonal blocks for single binary network, (a network with one relation whose socio matrix includes zero- one entries), which makes this algorithm be classified for community detection algorithm rather than block model problem. The other contribution based on structural equivalence, as an application of tab u-search algorithm, was presented by [12] for block modeling of a two-mode network, (a network with two sets of: actors and events whose data is recorded in the incidence matrix so that if an actor participate in an event, the corresponding element set as 1, otherwise set as 0), and followed with an adaptation of variable neighborhood search by [13] for finding ideal zero and one blocks, which is only applicable for binary network data. The greedy discrete PSO algorithm of [14] is the one of the latest contribution in partitioning networks which made PSO applicable, effective, and competitive in community detection problem for finding cohesive subgroups. Although PSO algorithm, originally was not designed for discrete valued problem, the greedy discrete PSO of [14] works well and produces high quality solutions for community detection problems in terms of modularity index of Girvan and Newman [15]. However this algorithm, as mentioned by authors, is not fit and parsimonious for large networks, because of high time consuming for its convergence. In addition, it is designed for community problem and is not suitable for block model problem.

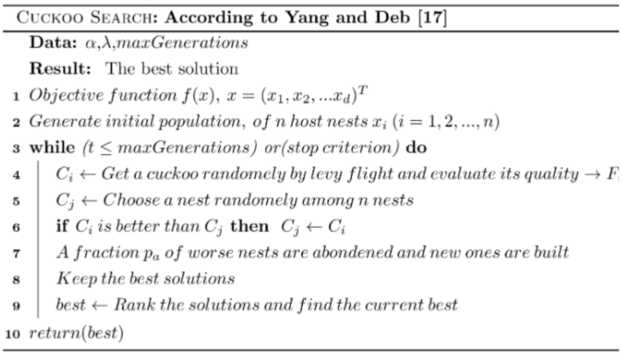
2.2 Cuckoo Search Algorithm (CS)

Metaheuristic methods usually are nature inspired algorithm and originally coordinate the interaction between local improvement procedures and procedures for avoiding local optimum and have capabilities to accomplish a robust search in a solution space. In fact, these methods employ strategies for conquering the trap of local optimality in complex solution spaces. Essentially, metaheuristics are approximate optimization methods which provide acceptable solutions in a reasonable time for solving hard and complex problems in science and engineering. There are two natural phenomena which is inspired by researchers during the development of metaheuristic algorithms as mentioned by [16]: the Darwinian evolution, and the social behavior of living animals and insects (e.g., birds, bees, ant colonies, fireflies). While evolutionary algorithms (EAs), as a species of these methods, refers to a family of algorithms which have been induced from the Darwinian nature evolution, swarm intelligence (SI) based algorithms are influenced from the social behavior of animals. Among evolutionary algorithms, the Genetic Algorithm (GA), as the most applicable and famous algorithm, is a probabilistic search algorithm that iteratively transforms a set (population) of objects (usually a fixed-length binary string), each with an associated fitness value, into a new population of offspring objects using the Darwinian principle of natural selection and operations that mimic naturally occurring genetic operations such as mating recombination (crossover) and mutation.

In the family of the swarm intelligence algorithms, the Cuckoo Search algorithm (CS) as the one of the latest nature inspired metaheuristics, was developed by Yang and Deb [17], and is inspired by the brood parasitism of some cuckoo species laying their eggs in the nests of other bird species. According to Yang and Deb [17] cuckoo have the aggressive reproduction strategy. The term "brood parasitism" means that some species of cuckoo lays their eggs in the nests of other host birds (often other species) and includes three types: intraspecific brood parasitism, cooperative breeding, and nest takeover. There is a possibility that a conflict between some host birds and the intruding cuckoos is occurred. That is if a host bird discovers the extrinsic eggs, the discard of these foreign eggs or leave the nest and builds a new nest elsewhere will be chosen by the host birds. To adopt this behavior of cuckoo reproduction in forming an algorithm, Yang and Deb [17] constructed three rules:

1. Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest.
2. The best nests with high quality eggs will be carried over to the next generations.
3. The number of available host nests is fixed and any egg laid by a cuckoo may be discovered by the host bird with a probability $p_a \in (0, 1)$. In this case, the host bird can either get rid of the egg, or simply abandon the nest and build a completely new one.

Yang and Deb [17] using these rules, constructed cuckoo search algorithm as follows



```
CUCKOO SEARCH: According to Yang and Deb [17]
Data: $\alpha$,$\lambda$,$maxGenerations$
Result: The best solution
Objective function $f(x)$, $x = (x_1, x_2, ...x_d)^T$
Generate initial population, of $n$ host nests $x_i$ $(i = 1, 2, ..., n)$
while $(t \leq maxGenerations)$ or(stop criterion) do
    $C_i \leftarrow$ Get a cuckoo randomely by levy flight and evaluate its quality $\rightarrow F$
    $C_j \leftarrow$ Choose a nest randomely among n nests
    if $C_i$ is better than $C_j$ then $C_j \leftarrow C_i$
    A fraction $p_a$ of worse nests are abondened and new ones are built
    Keep the best solutions
    best $\leftarrow$ Rank the solutions and find the current best
return(best)
```

Figure 1. Cuckoo Search Algorithm of Yang and Deb

Yang and Deb [17] demonstrated he superiority of cuckoo search over genetic algorithm and PSO through several standard tests. In all experiments, the solution space is continuous and results corroborate that cuckoo search clearly outperforms GA and PSO. The results reported by Yang and Deb [17], motivated us to employ cuckoo search to find the optimal blockmodel. Because PSO algorithm was dominated by GA in blockmodel problem, we adopt GA as a basis for comparison with cuckoo search.

3. Proposed Method

In this section, required definitions of our research is presented. The definition of the blockmodel problem is prepared at the first subsection. In addition, the specification of genetic algorithm and cuckoo search algorithm is prepared

3.1 Blockmodel Problem

As mentioned in the previous section, we search for the best structure agree with the original network data. This means that our goal is to find the structure which estimates relatively close to original network ties. The closeness to original network data can be measured by either statistically goodness of fit indices or deviance indices. This measurement, simply is done by comparison of original network data with estimated network data by a given blockmodel. Thus it is necessary to define the estimation of network ties under a given blockmodel. This estimation is defined in general form to be applied in wide range of network data and then it is revised to be applicable for calculation of the log likelihood ratio statistics, in the special case of discrete ordinal network data.

Suppose $\mathcal{N} = \{1,2,\ldots,n\}$ is a set of actors in network $\mathcal{S}$ and $\mathcal{R} = \{1,2,\ldots,R\}$ is a set of R valued relations measured on $\mathcal{N}$, and $\mathcal{X}_{\mathrm{r}}$ is the sociomatrix of rth relation where $\mathcal{X}_{\mathrm{rij}}$ is the strength of the tie from actor i to actor j on the rth relation. In addition, let $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_p\}$ be a set of positions and Φ be a mapping function, which assigns actors to positions; that is:

$$\begin{aligned} &\phi: \mathcal{N} \longrightarrow \mathcal{P} \\ &\forall\, i \in \mathcal{N}\ \exists!\ j \in \mathcal{P} \ni \phi(i) = j \end{aligned} \tag{1}$$

The relation between and within positions can be represented by density matrices. Suppose $\Delta = [\delta_{ruv}]_{r \times p \times p}$ is an array of density matrices among these p positions such that if n_k, n_l are respectively the number of members in positions k and l, we have:

$$\Delta_{rkl} = \begin{cases} \sum_{i \in \mathcal{P}_k} \sum_{j \in \mathcal{P}_l} x_{rij}/n_k n_l\,, & if\ k \neq l \\ \sum_{i \in \mathcal{P}_k} \sum_{i \in \mathcal{P}_k} x_{rij}/n_k(n_k - 1)\,, & if\ k = l \end{cases} \tag{2}$$

Now, the blockmodel $\mathcal{B}$ can be defined as: $\mathcal{B} = \langle \phi, \Delta \rangle$. The hypothetical nature of a blockmodel makes it capable to estimate the strength of each tie, and once a partition is established, either posteriori or a priori, the model can estimate relational ties. In the other words, we have:

$$E\left(x_{rij} \middle| \mathcal{B} = \langle \phi, \Delta \rangle\right) = \hat{x}_{rij}^{\mathcal{B}} = \Delta_{r\phi(i)\phi(j)} \tag{3}$$

In fact, $\hat{\boldsymbol{x}}_{\boldsymbol{rij}}^{\boldsymbol{\mathcal{B}}}$ is the expected strength of tie, from actor i to actor j in the rth relation under hypothesis $\mathcal{B}$.

In order to calculate goodness of fit of a given blockmodel, in terms of G^2, we need to assume strengths of ties is categorical. Suppose $\mathcal{R} = \{1,2,\ldots,R\}$ is a set of R ordinal valued relations measured on $\mathcal{N}$, and $\mathcal{X}_{\mathrm{r}}$ is the sociomatrix of rth relation where $\mathcal{X}_{\mathrm{rij}}$ is the strength of the tie from actor i to actor j, on the rth relation. In addition, assume that these strengths of ties come from a bounded ordinary set $\mathcal{V}_r$ which has C_r categories on the rth relation. Let ψ be a mapping function from $\mathcal{V}_r$ to $\{1,2,3,\dots,C_r\}$ ordinary set such that:

$$\forall\, v \in \mathcal{V}_r\ \exists!\ c \in \{1,2,\dots,C_r\} \ni \psi(v) = c \qquad (4)$$

Now, we define a four dimensional $(n \times n \times C_r \times C_r)$ cross-classified array $\boldsymbol{Y} = \{Y_{rijkl}\}$ as:

$$Y_{rijkl} = \begin{cases} 1 \ \ if \ \left(\psi(\mathcal{X}_{rij}), \psi(\mathcal{X}_{rji})\right) = (k,l)\ , where\ k,l \in \{1,2,\dots,C_r\} \\ 0 \qquad otherwise \end{cases} \qquad (5)$$

As noted by [18], the (i,j) th cell of Y is a $C_r \times C_r$ submatrix which has a single 1 in the (k, l) th cell and the remaining $C_r^2 - 1$ elements will be 0. Thus, these submatrices can be simply viewed as indicator matrices, giving the "state" of each $(\boldsymbol{\mathcal{X}_{rij}}, \boldsymbol{\mathcal{X}_{rji}})$ dyad. Now the estimation under blockmodel B for the state of $(\boldsymbol{\mathcal{X}_{rij}}, \boldsymbol{\mathcal{X}_{rji}})$ dyad is:

$$\hat{y}^{\mathcal{B}}_{rijkl} = Pr\left(y_{rijkl} = 1\right) = \begin{cases} \sum_{i\in\mathcal{P}_u} \sum_{j\in\mathcal{P}_v} Y_{rijkl}/(n_u n_v)\ , & u \neq v \\ \sum_{i\in\mathcal{P}_u} \sum_{j\in\mathcal{P}_u} Y_{rijkl}/n_u(n_u - 1)\ , & u = v \end{cases} \qquad (6)$$

Where counts n_u and n_v are the number of actors in positions $\mathcal{P}_u$ and, $\mathcal{P}_v$, respectively. Now, estimation of relational ties can be written as:

$$\hat{x}^{\mathcal{B}}_{rij} = \sum_{v\in\mathcal{V}_r} v\hat{y}^{\mathcal{B}}_{rijk+} = \sum_{v\in\mathcal{V}_r} v \sum_{l=1}^{C_r} \hat{y}^{\mathcal{B}}_{rijkl} \qquad (7)$$
$$where\ i \neq j, \quad \psi(v) = k, and\ \ l = 1,2,\dots,C_r$$

In order to assess how close a given blockmodel estimation is to original relational data, deviance indices such as sum of absolute error, sum of squared error (SSE), inconsistency index[5] and goodness of fit indices such as matrix correlation, ρ, and the log ratio statistics, G^2, can be used.

$$SAE = \delta_{x1} = \sum_{r,i,j} \left|x_{rij} - \hat{x}^{\mathcal{B}}_{rij}\right| \qquad (8)$$

$$SSE = \sum_{r,i,j} \left(x_{rij} - \hat{x}^{\mathcal{B}}_{rij}\right)^2 \qquad (9)$$

$$\rho = \left(\sum (x_{rij} - \bar{x}_r)\left(\hat{x}^{\mathcal{B}}_{rij} - \bar{x}_r\right)\right) \Big/ \left(\sqrt{\sum (x_{rij} - \bar{x}_r)^2}\ \sqrt{\sum \left(\hat{x}^{\mathcal{B}}_{rij} - \bar{x}_r\right)^2}\right) \qquad (10)$$

$$Inconsistency = \sum_{\phi(i),\phi(j)} \alpha \left\| x^{+}_{rij} \middle| \Delta_{r\phi(i)\phi(j)} < 0 \right\| + (1-\alpha) \left\| x^{-}_{rij} \middle| \Delta_{r\phi(i)\phi(j)} > 0 \right\| \qquad (11)$$

$$G^2_{\mathcal{B}} = 2 \sum_{r} \sum_{i<j} \sum_{k,l} y_{rijkl} log\left(y_{rijkl}/\hat{y}^{\mathcal{B}}_{rijkl}\right) \qquad (12)$$

Inconsistency index was defined by [5] as an average count of positive ties in negative blocks and negative ties in positive blocks and if there are no specific preferences between inconsistency in positive blocks and inconsistency in negative blocks then $\alpha = 0.5$. Except for G^2, the other measures are not used for statistical goodness of fit test statistics, so we called them as descriptive indices. While SAE and SSE are measures of dissimilarity, ρ is a measure of similarity between the two matrices, and so $(1 - \rho)$ can measure dissimilarity. That is, SAE, SSE and $(1 - \rho)$ can be viewed as indices of lack of fitness of a given blockmodel. These indices have the advantage of applicability in any one mode relational data such as dichotomous, signed, ordered valued and valued relation, while G^2/ inconsistency index is suitable for ordinal / signed relations. Now, blockmodeling problem can be formulated as:

$$\begin{gathered} Minimize\ Lack\ of\ Fit \\ subject\ to: \\ \forall\, u \in \mathcal{P}: n_u \geq 1 \\ Where\ Lack\ of\ Fit \in \{SAE, SSE, (1-\rho), G^2, Inconsistency\} \end{gathered} \qquad (13)$$

3.2 Genetic Algorithm for Blockmodel Problem (BMGA)

In all application of GA, at first, the chromosome structure must be defined. In blockmodeling problem, chromosome structure corresponds to partition vector; that is a chromosome string is a vector of length n such that the value of each element is the position (class) number of each actor. In fact, a partition vector corresponds to a chromosome (string), and each solution decodes as partition vector.

1	*2*	*3*		*n-2*	*n-1*	*n*
2	*4*	*1*	*....*	*3*	*8*	*7*

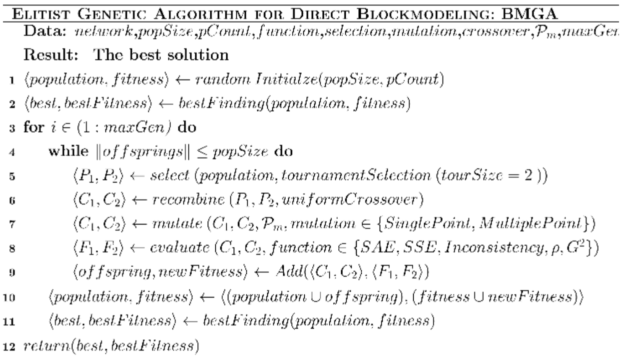
Figure 2. Partition vector as Chromosome

In the second step is recombination process of algorithm must be clarified. In GA recombination process consist of two major operators: crossover and mutation. Crossover operates on couple chromosomes; that is the mating process can be done between two chromosomes through crossover operators. Although there are several types of crossover: single point, multiple point, and uniform, we adopt uniform crossover in each run of algorithm in all experiments.

Another operator is mutation. Mutation operator increases the possibility of escape from local optimum and attempts to change the value of one or more genes of a given chromosome to a new randomly-generated value. Several types of mutation operators are proposed in the literature such as single point, double point, multiple point, swap, etc. Among various types of mutation, single point mutation is more common, which operates on a single chromosome, but the multiple point mutation is also used in our experiments

A selection method for choosing parent chromosomes to reproduce new child chromosomes is needed. Among various selection methods such as roulette wheel, tournament and rank, we used tournament methods with tour size equals to 2.

In order to make algorithm be converged we unify successive generations as new generation. Thus we have an elitist genetic algorithm which is shown in figure 3



```
ELITIST GENETIC ALGORITHM FOR DIRECT BLOCKMODELING: BMGA
Data: $network, popSize, pCount, function, selection, mutation, crossover, \mathcal{P}_m, maxGe$
Result: The best solution
$\langle population, fitness \rangle \leftarrow random\ Initialze(popSize, pCount)$
$\langle best, bestFitness \rangle \leftarrow bestFinding(population, fitness)$
for $i \in (1 : maxGen)$ do
    while $\|offsprings\| \leq popSize$ do
        $\langle P_1, P_2 \rangle \leftarrow select\ (population, tournamentSelection\ (tourSize = 2\ ))$
        $\langle C_1, C_2 \rangle \leftarrow recombine\ (P_1, P_2, uniformCrossover)$
        $\langle C_1, C_2 \rangle \leftarrow mutate\ (C_1, C_2, \mathcal{P}_m, mutation \in \{SinglePoint, MultiplePoint\})$
        $\langle F_1, F_2 \rangle \leftarrow evaluate\ (C_1, C_2, function \in \{SAE, SSE, Inconsistency, \rho, G^2\})$
        $\langle offspring, newFitness \rangle \leftarrow Add(\langle C_1, C_2 \rangle, \langle F_1, F_2 \rangle)$
    $\langle population, fitness \rangle \leftarrow \langle (population \cup offspring), (fitness \cup newFitness) \rangle$
    $\langle best, bestFitness \rangle \leftarrow bestFinding(population, fitness)$
$return(best, bestFitness)$
```

Figure 3.Elitist Genetic Algorithm for blockmodeling

3.3 Cuckoo Search for Blockmodel Problem (BMCS)

According to figure 1, a new cuckoo from existing cuckoo is recreated by levy flight random walk; that is for cuckoo i, new cuckoo is obtained by levy flight as:

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus Levy(\lambda) \tag{14}$$

Where α > 0 is the step size which is related to the scales of the given problem. As noted by Yang and Deb [17], in most cases, α = 1 is suitable step size. The equation (9) is essentially the stochastic equation for random walk which is in turn a Markov chain whose next status only depends on the current state (the first term in the above equation) and the transition probability (the second term). The operator ⊕ indicates entry-wise multiplications. Yang and Deb notes: "This entry-wise product is similar to those used in PSO, but the random walk via Levy flight is more efficient in exploring the search space as its step length is much longer in the long run". The Levy flight essentially provides a random walk while the random step length is drawn from a Levy distribution with infinitive mean and variance:

$$Levy \sim u = t^{-\lambda}, \qquad where\ 1 < \lambda \leq 3 \tag{15}$$

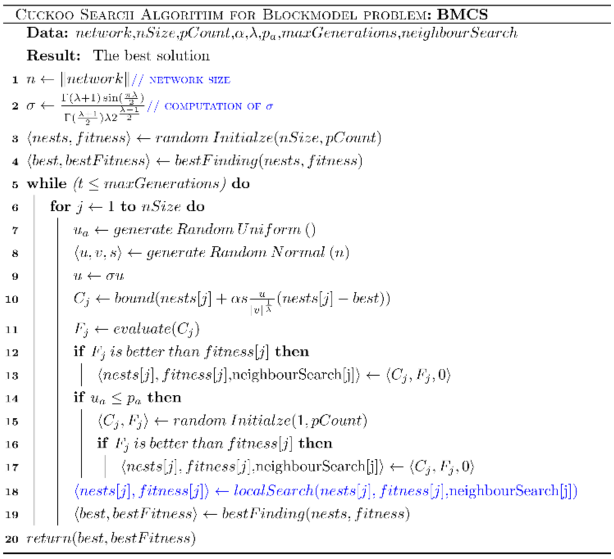
In order to draw a sample from Levy distribution, Yang [19] used Mantegna's algorithm which calculates a Levy distribution number as:

$$s \sim Levy : s = \frac{u}{|v|^{1/\lambda}} \ni u \sim N(0, \sigma^2)\ and\ v \sim N(0,1) \tag{16}$$

Where σ is obtained as:

$$\sigma = \left[\Gamma(1+\lambda) sin(\pi\lambda/2) / \left(\Gamma((1+\lambda)/2)\lambda 2^{((\lambda-1)/2)}\right)\right]^{1/\lambda} \tag{17}$$

In order to apply cuckoo search for blockmodeling problem, each nest/cuckoo corresponds to a partition vector. In fact in cuckoo search, nest/egg corresponds to chromosome/gen in genetic algorithm. In order to achieve a hybrid cuckoo search algorithm, we incorporate a local search procedure in the discovery of foreign eggs section. In fact, as perceived by figure 4, in line 18, for each solution being checked to be replaced by new randomly created solution, if the quality of new solution is worse than current under check solution, we perform a local search around this solution. In this way, the quality of solutions is not deteriorated and we implemented a modified cuckoo search algorithm for blockmodeling problem. The input parameter neighbourSearch is used to decide whether the local search is performed or not. That is if this parameter is set to zero the local search procedure of line 18, without any process, will keep the current cuckoo and BMCS will be a simple cuckoo search algorithm. The local search simply, searches for the best fitness among the first neighbors of a solution. To construct the first neighborhood for a partition vector, one point (position) randomly is selected and its value is changed. To prepare more exploration, more than one point (say two) can be used to be changed



```
CUCKOO SEARCH ALGORITHM FOR BLOCKMODEL PROBLEM: BMCS
Data: $network, nSize, pCount, \alpha, \lambda, p_a, maxGenerations, neighbourSearch$
Result: The best solution
1  $n \leftarrow \|network\|$ // NETWORK SIZE
2  $\sigma \leftarrow \frac{\Gamma(\lambda+1)\sin(\frac{\pi\lambda}{2})}{\Gamma(\frac{\lambda+1}{2})\lambda 2^{\frac{\lambda-1}{2}}}$ // COMPUTATION OF $\sigma$
3  $\langle nests, fitness \rangle \leftarrow random\ Initialze(nSize, pCount)$
4  $\langle best, bestFitness \rangle \leftarrow bestFinding(nests, fitness)$
5  while $(t \leq maxGenerations)$ do
6      for $j \leftarrow 1$ to $nSize$ do
7          $u_a \leftarrow generate\ Random\ Uniform\ ()$
8          $\langle u, v, s \rangle \leftarrow generate\ Random\ Normal\ (n)$
9          $u \leftarrow \sigma u$
10         $C_j \leftarrow bound(nests[j] + \alpha s \frac{u}{|v|^{\frac{1}{\lambda}}}(nests[j] - best))$
11         $F_j \leftarrow evaluate(C_j)$
12         if $F_j$ is better than $fitness[j]$ then
13             $\langle nests[j], fitness[j]$,neighbourSearch[j]$\rangle \leftarrow \langle C_j, F_j, 0 \rangle$
14         if $u_a \leq p_a$ then
15             $\langle C_j, F_j \rangle \leftarrow random\ Initialze(1, pCount)$
16             if $F_j$ is better than $fitness[j]$ then
17                 $\langle nests[j], fitness[j]$,neighbourSearch[j]$\rangle \leftarrow \langle C_j, F_j, 0 \rangle$
18         $\langle nests[j], fitness[j] \rangle \leftarrow localSearch(nests[j], fitness[j]$,neighbourSearch[j]$)$
19         $\langle best, bestFitness \rangle \leftarrow bestFinding(nests, fitness)$
20 $return(best, bestFitness)$
```

Figure 4. Cuckoo search algorithm for blockmodel problem

4. Experiments

In this section, the performance of the proposed hybrid cuckoo search algorithm in comparison with genetic algorithm/ greedy discrete PSO, for blockmodel/ community detection problem using some famous examples in social network analysis literature, is examined.

In this regard, we use three famous samples including Kapferer's Tailor Shop sample in [20], aggregated Sampson's data discussed by [5], and finally world trade in miscellaneous manufactures of metal 1994 discussed by [21] as a case of interval valued relation.

Now, in the next subsection the performance of BMCS against BMGA is investigated and then the capability of the proposed hybrid cuckoo search for community detection problem is compared with the greedy discrete particle swarm optimization and finally the effect of parameters such as iteration, nest size and incorporating the proposed local search are studied.

4.1 The comparison of BMCS and BMGA

Since BMCS with “neighbourSearch” greater than zero, is converted to a hybrid cuckoo search algorithm (denoted by BMHCS hereafter) which employs a local search procedure in addition to conventional update process, it is obvious that, in each update process, the execution time of this hybrid version is greater than execution time of the simple cuckoo search. Therefore it is fair that execution time of each update/recombination process of simple cuckoo search/BMGA, is restricted by corresponding update process of BMHCS. That is in each update process of BMHCS the elapsed time is measured as Thcs and then the update/recombination process of simple cuckoo search/BMGA is run until the execution time does not exceed this Thcs. Even though the total execution time of BMHCS, eventually is smaller than the total execution time of BMCS/BMGA, this is indispensable and in favor of BMGA/BMCS and if BMHCS is demonstrated to be outperformed BMCS/BMGA this favoritism will be completely insignificant. In all experiments the parameters of cuckoo search algorithm according to implemented MATLAB code proposed by Yang [19] were set to $\langle \alpha = 0.1 | \lambda = 1.5 | p_a = 0.25 \rangle$.

The local search of BMHCS were applied by setting neighbourSearch parameter to non-zero value which means that in each update process, if the fitness of a cuckoo is not improved the first neighborhood of that cuckoo will be sought. As mentioned before, this neighborhood is constructed by randomly selecting some eggs/gens and changing their value to all possible value.

Now the results of experiments in each sample are presented separately:

1) **Sampson Monastery Data.**

This sample is about the relations measured by Sampson [22] between monks. These relational data consist of four signed relations: like, esteem, influence and praise. In the literature, traditionally, positive and negative ties are recorded with separate matrices for each signed relation; specifically these signed relations are converted such that: like⟶(liking, disliking), esteem⟶(esteem, disesteem), influence⟶ (positive influence, negative influence) and praise⟶(praise, blame). Doreian and Mrvar [5] summed the like, esteem and influence relations to create a valued signed relation which is called as Doreian- Sampson. This data is ordinal and valued in range of [-9, 9]. Since Doreian- Sampson data is a singed ordinal valued relation, experiments are done using inconsistency as fitness function.

For comparison purpose, the experiments were set as follows: for nest/population size two levels of 30 and 50, and for maxGens three levels of 50, 100, and 200, were considered and for each pair of (nest size, maxGens), BMGA and BMHCS were run 15 times and results were summarized in table 1 and were shown in figure 5, where box plot of results of BMHCS/BMGA with color of green/purple through maxGen of 50, 100, and 200 was drawn respectively.

The local search of BMHCS, for each cuckoo which is not improved by usual update process, selects randomly one egg of that cuckoo and change its value to all possible values and then among these values, the value which prepares the best fitness, is chosen for that egg.

In the table 1, the third column represents the p-value of a hypothesis test of superiority of genetic algorithm over proposed hybrid cuckoo search algorithm, in terms of fitness function and convergence time, using the nonparametric Wilcoxon rank sum test. This superiority, means that fitness function/ convergence time of BMGA is better/ greater than BMHCS. The forth column prepares a confidence interval, which was computed using the nonparametric “Sign test”, at the level of 95% of results produced by BMGA / BMHCS.

nSize	maxGen	$H_0: BMGA \geqslant BMHCS$		95% *Confidence Interval*	
		Inconsistency	Time	BMGA	BMHCS
30	50	0.1831	0.1065	(10.5, 14.41)	(10.5,13.82)
30	100	0.2417	0.03723	(10.5 12.23)	(10.5, 11.0)
30	200	0.3388	0.7332	(10.5, 10.5)	(10.5, 10.5)
50	50	0.9406	0.00052	(10.5, 11)	(10.5,13.82)
50	100	0.03276	0.003499	(10.5, 11)	(10.5, 10.5)
50	200	0.008987	0.06237	(10.5, 13.88)	(10.5, 10.5)

Table 1. The results of comparison between BMGA and BMCS for Sampson-Doreian Data

According to results of table 1, while there is no enough strong evidence for the superiority of BMGA/BMHCS over BMHCS/BMGA for the nest size equal to 30, there is strong evidence for superiority of BMGA/BMHCS over BMHCS/BMGA for nest size equals to 50; so that genetic algorithm outperform the proposed hybrid cuckoo search at the level of 50 of maxGen; but for maxGen larger than 50, ie 100 and 200, the BMHCS clearly outperforms the BMGA. Therefore, running with the more iteration, makes BMHCS more effective than BMGA and so the possibility of achieving the better solution is increased by the large maxGens for BMHCS. However except for nest size of 30 with maxGen of 200, the convergence time of BMGA is significantly lower than that of BMHCS

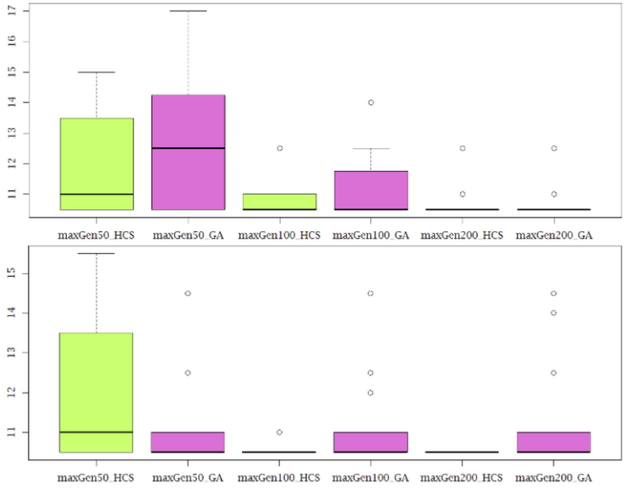


Figure 5. The boxplots of results produced by BMGA and BMHCS; Upper/Lower: the results of experiments with nest Size=30/50

Although, at the level of 200 for maxGen, it seems that the confidence interval of inconsistency index, produced by BMGA, for nest Size of 50 is greater than for nest Size of 30, statistically there is no significant differences between results produced by BMGA for maxGen of 200 at two levels of 30 and 50 for nest size.

2) **Kapferer's Tailor Shop**

The second sample is Kapferer's Tailor shop, discussed by [20]. This network has a size of 39×39 and two/four actors with indegree/outdegree equal to zero. Since this relational data is dichotomous, all mentioned indices, except for inconsistency, are applicable. Among them, SAE and G^2 as objective functions to minimize lack of fitness of the blockmodel problem were chosen.

The experiments and the local search were set as previous sample and results for G^2 and SAE were presented in table 2/ figure 6 and table 3/figure 7 respectively.

nSize	maxGen	$H_0: BMGA \geqslant BMHCS$		$95\%\ Confidence\ Interval$	
		G^2	Time	BMGA	BMHCS
30	50	1.00	0.2528	(572.25,577.91)	(588.47,593.5)
30	100	0.7558	0.000695	(572.25,579.06)	(572.62, 576.3)
30	200	0.01233	1.67e-05	(572.25,575.97)	(572.25,572.25)
50	50	1.00	0.009771	(572.25,575.97)	(585.38,595.21)
50	100	0.8696	3.39e-05	(572.25,575.31)	(572.25,576.24)
50	200	0.0398	2.85e-05	(572.25,572.25)	(572.25,572.25)

Table 2. The results of comparison between BMGA and BMCS for Kapferer Tailor's shop data in terms of G^2

In table 2, in third column the p-value of hypothesis test of the superiority of the genetic algorithm over the proposed hybrid cuckoo search, in terms of fitness function and convergence time, was computed. According to the table 2, as perceived from figure 6, while the superiority of BMGA over BMHCS for maxGen of 50 and for maxGen of 100 with nest size of 50, is significant, for maxGen equals to 200, the BMHCS clearly outperforms BMGA.

On the other hand, the convergence time of BMGA is significantly less than that of BMHCS, which makes BMGA be advantageous over BMHCS. However the performance of BMHCS, in finding high quality solutions for large maxGen justifies the high convergence time

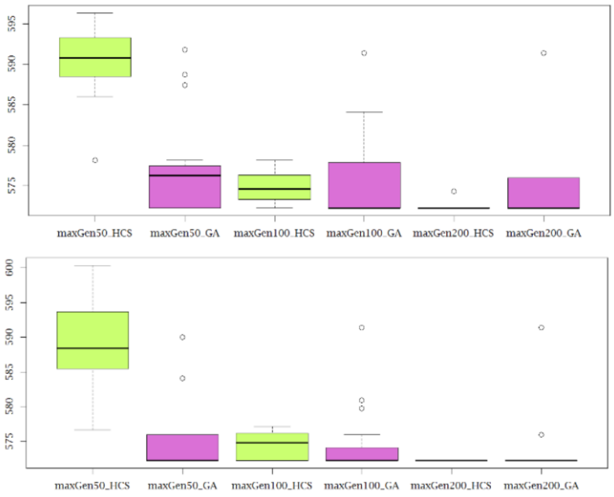


Figure 6. The boxplots of resulted G^2 produced by BMGA and BMHCS; Upper/Lower: the results of experiments with nest Size=30/50

The results of experiments in table 3, show that the genetic algorithm is not dominated by hybrid cuckoo search algorithm in minimizing SAE function. In addition, for each size of 30 and 50, as the maxGen increases, the quality of results found by BMHCS, is improved and eventually BMHCS finds the best solution which is obtained by BMGA. This means that the solutions found by BMHCS has the same quality as the solution obtained by BMGA. The quality of solutions of two algorithm can be compared from the boxplot presented in figure 7. In figure 7, it is obvious that the quality of solutions found by BMHCS, is improved by increasing maxGen so that there are no clear differences between BMHCS and BMGA for maxGen of 200. However as mentioned for experiments of G^2, the convergence time is the significant advantage of BMGA over BMHCS for experiments which use SAE as fitness function.

nSize	maxGen	$H_0: BMGA \geqslant BMHCS$		$95\%\ Confidence\ Interval$	
		Inconsistency	Time	BMGA	BMHCS
30	50	1.00	0.8779	(205.14,209.11)	(213.7, 217.85)
30	100	1.00	0.00052	(205.14,205.14)	(206.82,209.10)
30	200	0.7114	2.39e-05	(205.14,205.14)	(205.14,205.14)
50	50	1.00	0.8779	(205.14,209.11)	(213.7, 217.85)
50	100	1.00	1.67e-06	(205.14,205.14)	(206.37,207.91)
50	200	1.00	1.67e-06	(205.14,205.14)	(205.14,205.14)

Table 3. The results of comparison between BMGA and BMCS for Kapferer Tailor's shop data in terms of SAE

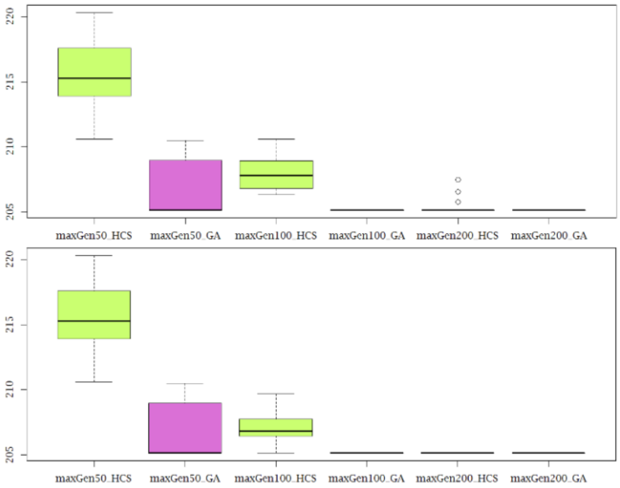


Figure 7.The boxplots of resulted SAE produced by BMGA and BMHCS; Upper/Lower: the results of experiments with nest Size=30/50

3) **World trade in miscellaneous manufactures of metal**

This data include miscellaneous manufactures of metal trade data among 80 countries in 1994. All countries with entries in the paper version of the Commodity Trade Statistics published by the United Nations were included, but for some countries, the 1993 data (Austria, Seychelles, Bangladesh, Croatia, and Barbados) or 1995 data (South Africa and Ecuador) were used because they were not available for 1994. Countries which are not sovereign are excluded because additional economic data were not available: Faeroe Islands and Greenland, which belong to Denmark and Macau (Portugal). Most missing countries are located in central

Africa and the Middle East, or belong to the former USSR. The arcs represent imports by one country from another for the class of commodities designated as '*miscellaneous manufactures of metal*', which represents high technology products or heavy manufacture. The absolute value of imports (in 1,000 US$) is used, but imports with values less than 1% of the country's total imports were excluded.

The data were used as a sample for the blockmodeling case in [21], and a core- peripheral structure as a result of conventional blockmodeling was demonstrated.

Since this relational data is interval valued, the application of G^2, as a measure of lack of fit, is neither appropriate nor practical; thus, in this case, the other indices are practical. In order to find the partition vector of most fitness with original data, matrix correlation, □, was used, and for this function, experiments were done separately.

In order to compare the performance of two algorithms, the configuration of experiments were set as follows: three levels of 20, 30, and 50 for nest/population size, and three levels of 40, 200, and 1000 for iteration or maxGen, were allocated. The local search of BMHCS as previous sample were set by neighbourSearch equals to 1, which constructs the first neighbor of current unimproved cuckoo by changing the value of one randomly selected egg of that cuckoo to all possible value and then adopts the best neighbor to be replaced by that cuckoo. All experiments were run 15 times and the results were summarized in table 4.

nSize	maxGen	$H_0: BMGA \geqslant BMHCS$		$95\%\ Confidence\ Interval$	
		Correlation:ρ	Time	BMGA	BMHCS
20	40	1.00	1.00	(0.472,0.592)	(0.319, 0.368)
20	200	1.00	0.000923	(0.645, 0.645)	(0.620, 0.6426)
20	1000	0.0189	8.01e-06	(0.645, 0.645)	(0.645, 0.686)
30	40	1.00	1.00	(0.613, 0.64)	(0.345, 0.383)
30	200	0.9994	1.7e-06	(0.645, 0.645)	(0.638, 0.645)
30	1000	0.001627	1.67e-06	(0.645, 0.645)	(0.645, 0.695)
50	40	1.00	0.9166	(0.636, 0.645)	(0.377, 0.417)
50	200	0.5191	1.66e-06	(0.645, 0.645)	(0.645, 0.645)
50	1000	0.0006402	1.69e-06	(0.645, 0.645)	(0.645, 0.695)

Table 4. The results of comparison between BMGA and BMCS for metal trade network data in terms of matrix correlation

While, for maxGen of 40 regardless of nest size, the quality of results returned by BMGA is better than those of BMHCS, the convergence time of the genetic algorithm is larger than that of the hybrid cuckoo search algorithm; that is for small iteration, such as 40, the dominance of BMGA over BMHCS coincides with the larger convergence time of BMGA than that of BMHCS. On the other hand, for the maxGen of 200 and 1000, with regard to the convergence time, the superiority of BMGA over BMHCS is significant. However, as perceived from figure 8, this superiority for maxGen of 1000 concurs with its inferiority to BMHCS with respect to the quality of solutions. An important note is that by increasing the number of iterations, BMGA falls into local optima and premature convergence.

4.2 The Hybrid Cuckoo Search vs GDPSO

In addition to blockmodel problem, the proposed hybrid cuckoo search (HCS) of figure 5, using modularity index as fitness function can be used for community detection problem. Therefore the performance of the hybrid cuckoo search can be compared with the greedy discrete PSO (GDPSO). Since GDPSO is a time consuming algorithm, the hybrid cuckoo search was run in the time restriction equal to execution time of GDPSO; that is in order to make fair comparison of the hybrid cuckoo search with GDPSO, in each iteration, the time of update process of GDPSO is saved as TGDPSO and then the update process of the hybrid cuckoo search is run until its execution time does not exceed TGDPSO. Thus is in all experiments, instead of setting iterations number, the execution time threshold equals to GDPSO running time, for comparison of HCS with GDPSO, were used. The experiments for all three samples of karate, dolphins, and football, which are discussed by the authors of GDPSO, were set as follows:

$$\langle c_1 = c_2 = 1.4961 | \omega = 0.7298 | popsize = 50 | gmax = 50 \rangle$$

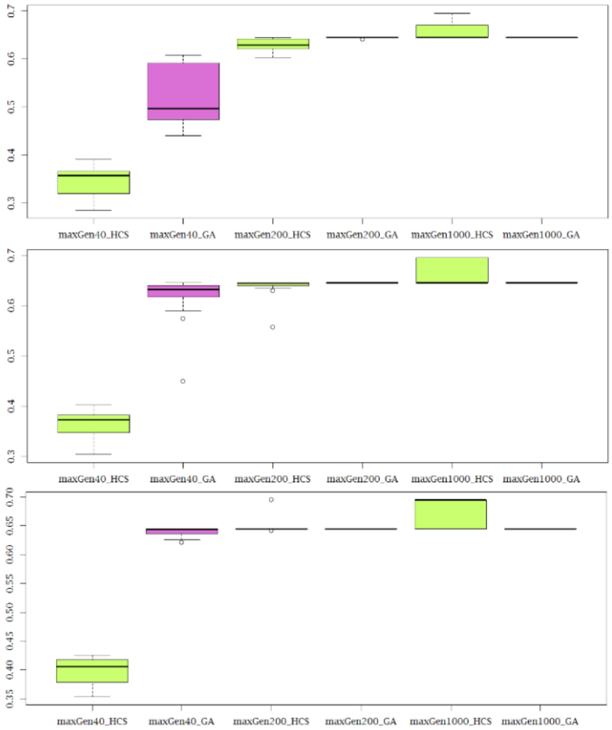


Figure 8. The box plot of results emerged from running BMHCS and BMGA respectively from up to down: for nest size of 20, 30, and 50

This configuration for c_1, c_2 and ω are the same as those used by the authors of GDPSO and the level of 50 for population size and gmax has been used by authors too. The values parameters of the HCS were equal to: $\langle \alpha = 0.1 | \lambda = 1.5 | p_a = 0.25 | \text{neighbourSearch} = 2 \rangle$. The value of 2 for neighbourSearch parameter means that for unimproved cuckoo during usual update process of cuckoo search, randomly two eggs are selected and then in the neighborhood emerged from changing the value of these two eggs to all possible values, the egg with the best fitness is selected. For the evaluation of the performance differences between two algorithms, the p- values of the superiority of GDPSO over HCS using Wilcoxon rank sum, based on 30 times running of two algorithms were computed. The results of this comparison for three samples, were summarized in table 5 and illustrated by box plots in figure 9.

#	Sample Name	Network size	nSize	maxGen	Q_{Avg}^{MCS}	Q_{Avg}^{GDPSO}	$H_0: GDPSO \geqslant HCS$ $p - value$:
1.	Karate	34×34	50	50	0.4198	0.4168	0.005515
2.	Dolphins	62×62	50	50	0.5279	0.525	3.4e-06
3.	Football	115×115	50	50	0.5830	0.5689	6.4e-07

Table 5. The comparison of the hybrid cuckoo search with GDPSO

As perceived from figure 9, the superiority of GDPSO over the hybrid cuckoo search (HCS) is clearly rejected in all three samples and thus results of the proposed hybrid cuckoo search algorithm is not dominated by results of GDPSO in community detection problem. Therefore the hybrid cuckoo search has enough quality to search the best solution in comparison with the GDPSO, so that the adoption of HCS to search the best partition vector for community detection problem as well as blockmodel problem, is reasonable and tenable.

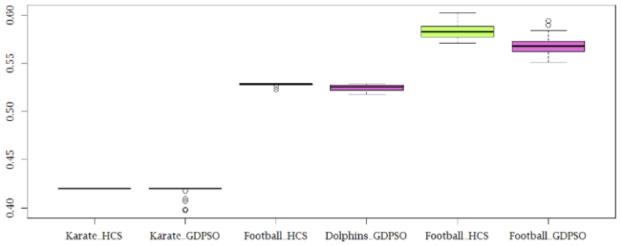


Figure 9. The boxplot of modularity index returned by HCS and GDPSO

4.3 The Effect of Parameters: maxGen, nestSize and algorithm

In all of the experiments mentioned before, the values of parameters used to run BMCS were not constant and for nestSize and maxGen several levels were considered. In addition to these parameters, whether the local search procedure is incorporated or not, must be investigated. For the study of effect of local search procedure, for all experiments reported in subsection 4-1, the results of running simple cuckoo search, denoted by CS, along with the results of the hybrid cuckoo search, denoted by HCS, were considered as a variable called algorithm. Therefor three factor of maxGen, nestSize and algorithm are studied for finding the important parameter set of BMCS. In order to study the effects of these parameters, for each sample of 4-1, interaction plot of factors was drawn and then three factor robust ANOVA test of Wilcox was performed separately.

According to interaction plot of figure 10 for Sampson-Doreian drawn for inconsistency index, it seems that the larger values of maxGen and nestSize, produces the lower inconsistency and in comparison with CS, HCS prepares the lower inconsistency. Furthermore, parallel/nonparallel curve in interaction plot of nestSize:maxGen/ nestSize:algorithm and maxGen:algorithm, seems to suggest that there are interaction effect between nestSize/ maxGen and algorithm and no interaction effect between nestSize and maxGen. This suggestion are corroborated with three-way robust ANOVA on the trimmed means ([23]), computed by t3way command of “WRS2” R package version: 0.9-1 and reported in table 6.

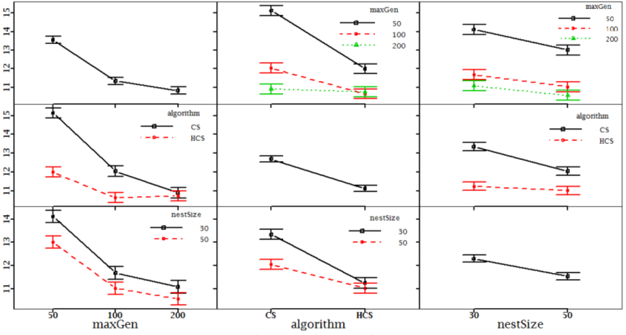


Figure 10. Interaction plot of three factor for Doreian-Sampson data in terms of Inconsistency index

The interaction plot of parameters for Kapferer’s Tailor shop samples in terms of G^2 and SAE were drawn in figure 11 and 12, respectively. According to these figures, it seems that there are the main effects corresponding to maxGen and nestSize and the interaction effects between algorithm and maxGen/nestSize. This can be examined by three factor robust ANOVA test summarized in table 7.

Effect	Value	p-value
maxGen	73.150	0.0001
algorithm	33.916	0.0001
nestSize	7.210	0.01
maxGen:algorithm	29.587	0.001
maxGen:nestSize	3.147	0.23
algorithm:nestSize	6.130	0.017
maxGen:algorithm:nestSize	4.703	0.117

Table 6. The results of three factor robust ANOVA test for Inconsistency as response variable in Sampson-Doreian sample

The results of table 7, reveals relatively different behavior of parameters for G^2 and SAE. While maxGen and nestSize are significant factors for both fitness function of G^2 and SAE, the algorithm is the main effect only for SAE. This indicates that CS and HCS are totally different/indifferent in minimizing SAE/G^2 so that CS produces better/ not better solutions than HCS in decreasing SAE/G^2. The interaction of maxGen and algorithm/nestSize is clearly significant for SAE (p-value=0.047/0.001), but relatively significant for G^2 (p-value=0.116/0.126). This makes both fitness functions be better improved by HCS rather than CS in maxGen of 200. Furthermore the interaction of three factors is significant only for SAE. Even though the effect of algorithm and its interaction with the other factors is not clearly significant for G^2, it is not reasonable that the interaction of algorithm with maxGen/nestSize be considered completely insignificant.

The final sample, was studied at the two levels of 20 and 30 for nestSize, and at the four levels of 40,200, 1000, and 1500 for maxGen. The interaction plot of figure

13, proposes the clear improvement in high levels of maxGen/nestSize and relatively indifferent behavior between CS and HCS. In addition the interaction/no interaction effect between algorithm and maxGen/nestSize is perceived. The results of three factor robust ANOVA test in table 8, reports the significant main effects of maxGen, algorithm, and nestSize. In addition the interaction effect between maxGen and algorithm along with interaction effect of maxGen, algorithm and nestSize, are significant. Since maxGen and and nestSize have positive effects on matrix correlation, these significant interaction effects, make HCS outperforms CS in large maxGen of 1000 and 1500 and nestSize of 30.

In all experiments, except for one experiment for G^2, the main effects of maxGen, algorithm, and nestSize are considered significant. Among these effects the maxGen seems be more prominent; that is while nestSize and algorithm are important too, the clear significant improvement of fitness function in all samples is concurred with increasing the value of maxGen. In addition, interaction effect of maxGen with algorithm in all samples for each fitness function is significant. This interaction effect cause HCS to be superior to CS in large maxGen. In fact by increasing the maxGen, the possibility that HCS is dominated by CS is decreased and eventually the HCS dominates CS in a large maxGen. This large maxGen for Sampson-Doreian, Kapfere Tailor shop, and metal trade network equals to 50, 200, and 1000 respectively. However in small maxGen, HCS is not clear advantage over CS.

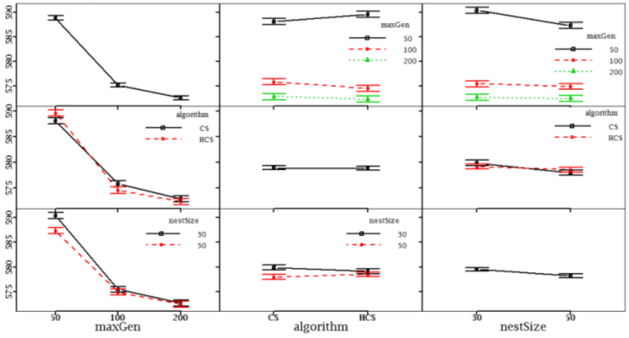


Figure 11. Interaction plot of three factor for Kapferer's Tailor shop data in terms of G^2

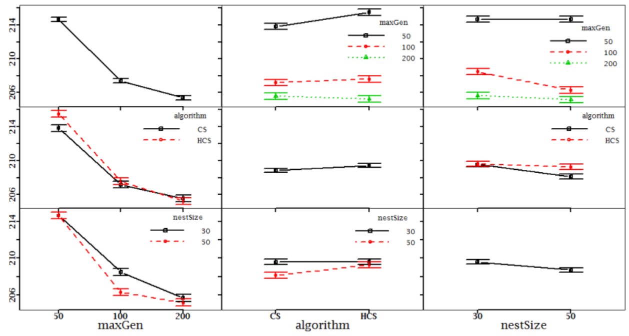


Figure 12. Interaction plot of three factor for Kapferer's Tailor shop data in terms of SAE

Effects	G^2		SAE	
	Value	p.value	Value	p.value
maxGen	579.148	0.000	525.020	0.000
algorithm	0.043	0.840	5.802	0.020
nestSize	5.026	0.031	5.377	0.025
maxGen:algorithm	4.639	0.116	6.736	0.047
maxGen:nestSize	4.444	0.126	17.070	0.001
algorithm:nestSize	0.347	0.560	1.992	0.165
maxGen:algorithm:nestSize	0.334	0.850	5.401	0.083

Table 7. The results of three factor robust ANOVA test for G^2/SAE as response variable in Kapferer Tailor shop sample

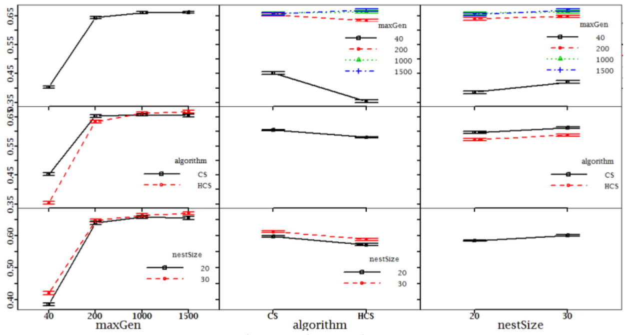


Figure 13. Interaction plot of three factor for metal trade network data in terms of Correlation

Effects	Value	p.value
maxGen	1849.138	0.0001
Algorithm	21.691	0.0001
nestSize	19.98	0.001
maxGen:algorithm	95.840	0.001
maxGen:nestSize	7.237	0.09
algorithm:nestSize	1.122	0.294
maxGen:algorithm:nestSize	10.194	0.031

Table 8. The results of three factor robust ANOVA test for matrix correlation as response variable in metal trade network

5. Conclusion

In this paper, we defined the blockmodel problem as an optimization problem and several fitness functions to measure how a given blockmodel presents a social structure agree with the original network data, were considered. Then we proposed a hybrid cuckoo search algorithm for solving blockmodel problem. This algorithm in comparison with an elitist genetic algorithm, through several samples, was not dominated and specifically for relatively large iteration, the proposed cuckoo search outperforms genetic algorithm. In addition to blockmodel problem, the performance of algorithm was studied for community detection problem and the quality of solutions produced by the algorithm clearly dominates the quality of those returned by its rival greedy discrete PSO algorithm. The effects of parameters were examined using non-parametric statistical tests and the results of these tests indicate that the large iteration plays important role for making clear improvement in fitness function for cuckoo search algorithm.

In addition significant interaction effect between iteration and incorporating the local search procedure, makes the proposed hybrid cuckoo search be advantageous over the simple cuckoo search algorithm. In fact for a large iteration the hybrid cuckoo search improves the fitness function better than the simple cuckoo search algorithm.

However the value of this large iteration was not known as a fixed and constant number and it must be found for each network data separately.

Although the role of local search was known significant for large iterations in this study, it is important to incorporate more efficient local search approach to achieve better results in relatively low execution time. This efficient hybridization of local search and cuckoo search algorithm is the main issue to be followed in future works.

References

[1] L. C. Freeman, "The development of social network analysis-With an emphasis on recent events," The sage handbook of social network analysis, vol. 21, pp. 26-39, 2011.

[2] F. Lorrain and H. C. White, "Structural equivalence of individuals in social networks," The Journal of mathematical sociology, vol. 1, pp. 49-80, 1971.

[3] P. Doreian, V. Batagelj, and A. Ferligoj, Generalized blockmodeling vol. 25: Cambridge university press, 2005.

[4] P. Doreian, V. Batagelj, and A. Ferligoj, "Generalized blockmodeling of two-mode network data," Social networks, vol. 26, pp. 29-53, 2004.

[5] P. Doreian and A. Mrvar, "Partitioning signed social networks," Social Networks, vol. 31, pp. 1-11, 2009.

[6] A. Žiberna, "Generalized blockmodeling of sparse networks," Metodoloski zvezki, vol. 10, p. 99, 2013.

[7] A. Žiberna, "Blockmodeling of multilevel networks," Social Networks, vol. 39, pp. 46-61, 2014.

[8] A. Žiberna, "Generalized blockmodeling of valued networks," Social networks, vol. 29, pp. 105-126, 2007.

[9] T. James, E. Brown, and C. T. Ragsdale, "Grouping genetic algorithm for the blockmodel problem," IEEE Transactions on Evolutionary Computation, vol. 14, pp. 103-111, 2010.

[10] O. C. Herfindahl, "Concentration in the steel industry," Columbia University., 1950.

[11] A. O. Hirschman, "The paternity of an index," The American Economic Review, vol. 54, pp. 761-762, 1964.

[12] M. Brusco and D. Steinley, "A tabu-search heuristic for deterministic two-mode blockmodeling of binary network matrices," Psychometrika, vol. 76, pp. 612-633, 2011.

[13] M. Brusco, P. Doreian, P. Lloyd, and D. Steinley, "A variable neighborhood search method for a two-mode blockmodeling problem in social network analysis," Network Science, vol. 1, pp. 191-212, 2013.

[14] Q. Cai, M. Gong, L. Ma, S. Ruan, F. Yuan, and L. Jiao, "Greedy discrete particle swarm optimization for large-scale social network clustering," Information Sciences, vol. 316, pp. 503-516, 2015.

[15] M. Girvan and M. E. Newman, "Community structure in social and biological networks," Proceedings of the national academy of sciences, vol. 99, pp. 7821-7826, 2002.

[16] I. Fister, D. Strnad, X.-S. Yang, and I. Fister Jr, "Adaptation and hybridization in nature-inspired algorithms," in Adaptation and Hybridization in Computational Intelligence, ed: Springer, 2015, pp. 3-50.

[17] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on, 2009, pp. 210-214.

[18] S. Wasserman and K. Faust, Social network analysis: Methods and applications vol. 8: Cambridge university press, 1994.

[19] X.-S. Yang, Nature-inspired metaheuristic algorithms: Luniver press, 2010.

[20] K. Nowicki and T. A. B. Snijders, "Estimation and prediction for stochastic blockstructures," Journal of the American Statistical Association, vol. 96, pp. 1077-1087, 2001.

[21] W. de Nooy, A. Mrvar, and V. Batagelj, Exploratory Social Network Analysis with Pajek vol. 27: Cambridge University Press, 2005.

[22] S. Sampson, "Crisis in a cloister, unpublished Ph. D," Dissertation, Dept. of Sociology, Cornell University, USA, 1969.

[23] R. R. Wilcox, Introduction to robust estimation and hypothesis testing: Academic Press, 2011.

Saeed Nasehimoghaddam graduated at BSc and MSc levels from faculty of industrial engineering, in Iran University of Science and Technology (IUST) respectively in 2005 and 2010. His focus during graduate studies was on socio economic systems engineering. After graduation from IUST at MSc level he followed his scientific interest in the University of Zanjan as a lecturer at engineering faculty and taught some relevant courses: such as Layout Planning, Engineering Economy, Engineering Statistics, Social Network Analysis, Project Planning and Control. His main research area is related to Data Mining and Social Network Analysis, and Computational Intelligence.

Mehdi Ghazanfari He received the BSc degree in industrial engineering from Iran University of Science and Technology (IUST), Iran, in 1986, and the MSc degree in system engineering from the Amirkabir University of Technology (AUT) Tehran, Iran, in 1989 and the PhD degree in industrial engineering from the University of New South Wales (UNSW), Sydney, Australia in 1994, respectively. he was an Assistant Professor, became an Associate Professor in 2000, and a Professor in 2003 and so far he has held lecturing position at the faculty of industrial engineering of IUST

Babak Teimourpour *O*btained his MSc degree in Socio-Economic Systems Engineering from Institute for Research on Planning and Development, Tehran, and he received his PhD in Industrial Engineering from Department of Industrial Engineering, Tarbiat Modares University (TMU), Tehran. He teaches PhD and MS level courses. His teaching interests include 'Data Mining' and 'Social Network Analysis'.