# Checking the Reliability of a Linear-Programming based Approach towards Detecting Community Structures in Networks [1]

William Y. C. Chen[2], Andreas W. M. Dress[3], and Winking Q. Yu[4]

[2,4]Center for Combinatorics, LPMC, Nankai University, Tianjin 300071, P.R. China

[3]CAS-MPG Partner Institute for Computational Biology, Shanghai Institutes for Biological Sciences, Chinese Academy of Sciences, Shanghai 200031, P.R. China

[3]Max-Planck-Institut fuer Mathematik in den Naturwissenschaften, Inselstrasse 22-26, D-04103 Leipzig, Germany

[2]chen@nankai.edu.cn, [3]dress@sibs.ac.cn, [4]yuqiang@mail.nankai.edu.cn

**Abstract.** We investigate the reliability of a recent approach to use parameterized linear programming for detecting community structures in networks. Using a one-parameter family of objective functions, a number of "perturbation experiments" document that our approach works rather well. We also analyze a real-life network and a family of benchmark networks.

## 1 Introduction

A network is viewed as a simple graph $G = (V, E)$ with vertex set $V$ and edge set $E \subseteq \binom{V}{2}$ ($\binom{V}{2}$ denoting the set of all 2-subsets of $V$). It has been proposed in [7] to employ linear programming (LP) to optimally approximate $G$ (by inserting and deleting edges) by a *community network* $G' = (V, E')$, i.e., a graph $G'$ that is a disjoint union of complete subgraphs.

More precisely, presupposing that any changes of $G$ should be penalized by summing up over penalties to be paid for single edge deletion and insertion, it was proposed in [7] to introduce a one-parameter family $\big(LP_G(s) : s \geq 1\big)$ of LP problems

- defined by constraints ensuring that the desired community networks all correspond, in a one-to-one fashion, to the feasible $\{0, 1\}$-solutions of each of these LP problems,
- and indexed by a parameter $s$ that is used for automatically calibrating the penalty to be paid for deleting an edge.

[1]This paper is a postprint of a paper submitted to and accepted for publication in "IET Systems Biology" and is subject to Institution of Engineering and Technology Copyright. The copy of record is available at IET Digital Library

It was shown that, for all large values of $s$, the LP problem $LP_G(s)$ has a unique solution that, in addition, is integral and corresponds to that community network $\overline{G} = (V, \overline{E})$ in which two vertices are connected by an edge if and only if they are contained in the same connected component of $G$: If deleting becomes too expensive, only inserting — and never deleting — edges presents the most economical solution.

Much more surprising, however, was the observation that community networks which researchers considered to representing very good, if not "the best" approximation of $G$ were almost invariably found for a very specific value of $s$ that could be characterized in a simple, purely arithmetic fashion: The smallest value $s^* = s^*(G)$ among all $s \geq 1$ for which the corresponding problem $LP_G(s^*)$ has, at least, one integral solution.

The main results of this paper are presented in Section 4: By a series of experiments, we investigate the reliability of the LP-based algorithm for community-structure detection proposed in [7] that we recall in Section 3. Before this, we shortly review some current community-structure studies in the next section. We also analyze another series of increasingly more difficult (and much studied) artificial benchmark cases in Section 5 and — last, but not least — one *real-life* example in Section 6. All these experiments document that the LP-based approach can well compete in quality — though not yet in speed — with more established heuristics and, thus, demonstrate the need of a new algorithm that incorporates the virtues of both types of approaches, precision and speed, and thus may outperform all of them.

# 2 Networks and Community Structures

Networks are *snapshots of dynamical systems*, and dynamical systems are *networks in action* [32, 37]. Consequently, there is some good hope that proper network analysis can help to elucidate the dynamics of relevant systems — from the World-Wide Web [1, 22], scientific collaborations, and citation networks [24] to the life sciences, e.g., the ecological, regulatory, protein, and metabolic networks [10, 19, 20, 34].

To apply *standard* methods of network analysis, a lot of detailed input information about the mechanisms of interactions between the various agents participating in the network's activity is required. Given such information, a lot of detailed information about its dynamics can be deduced by solving the resulting differential equations and/or computer simulation.

However, this approach has serious limitations: In many networks of interest in biology, such input information is simply not available. So the question arises: What can be done if all that is known are the network's agents represented by a collection $V = V(G)$ of *vertices* of the network $G$, and the network's *topology*, i.e., the subset $E = E(G)$ of the set $\binom{V}{2}$ consisting of those pairs $\{u, v\} \in \binom{V}{2}$ of distinct agents $u, v$ (also called the *edges* of the network) that we believe to strongly interact with each other?

Attempts to addressing such questions have received much attention ever since the current network hype began with the proclamation of *scale-free* [5] and *small-world* [37] (see also [2]) networks as constituting important new and universally applicable paradigms of interaction schemes observed in real-world systems, and suggesting fundamentally new basic laws governing important processes studied in the natural and the social sciences.

What we are concerned with here is one currently quite popular proposal within this program, i.e., the proposal of using the network's topology for deriving its *community structure*, that is, for grouping the network's agents into disjoint *communities* consisting of agents that appear to strongly interact with each other and not so strongly with those in the other communities: See [12, 23, 26, 29, 36] for a discussion of relevant definitions.

Furthermore, many approaches to *detect* communities in networks have been developed over the years, from spectral bisection [28], the Kernighan-Lin algorithm [21], and *hierarchical clustering* [13] to Girvan and Newman's landmark paper [14] that inspired much further work (see for instance [29, 33, 38]). In [26], Newman and Girvan proposed a quantitative measure dubbed *modularity* to compare the appropriateness of distinct community structures constructed for a given fixed network and developed an algorithm searching for *modularity-optimal* community structures (cf. [25]). Their approach was further improved by the "CNM algorithm" proposed by Clauset et al. [8] who developed an extremely fast heuristic that greedily searches for modularity-optimal community structures, returns demonstrably good results for many real-world networks, and has often been used successfully in recent years (see, e.g., [35] for a recent biological application).

Furthermore, apparently quite unaware of (a) the work of Grötschel and Wakabayashi [15, 16] and (b) the relationship between their own work and community-structure detection, Demaine and Immorlica [11] proposed to employ LP procedures for dealing with a (generalized version of) the "correlation clustering" problem studied by Bansal, Blum, and Chawla [4] that – at a first glance – looks very similar to our approach. However, their work does not only have a distinct (though related) goal, our method also employs a rather different strategy of using LP for finding (hopefully) relevant solutions. Yet, as Demaine and Immorlica were concerned with the algorithmic aspects of solving just single integer LP problems by some rounding technique, we do hope that the speed of our method can be improved (on the expense of accuracy) by incorporating their ideas into our procedure.

In the present paper, however, we will restrict our attention exclusively to investigating the reliability of our own approach as proposed in [7].

# 3 The Basic Set-Up

Let us now recall the notations, definitions, and results from [7]: Given a finite set $V$, consider

- the $\mathbb{R}$-vectorspace $\mathbb{R}^{\binom{V}{2}}$ consisting of all maps

$$x : \binom{V}{2} \to \mathbb{R} : \{u, v\} \mapsto x(u, v)$$

from $\binom{V}{2}$ into $\mathbb{R}$, the real-number field,

- the hypercube $[0, 1]^{\binom{V}{2}} \subset \mathbb{R}^{\binom{V}{2}}$ consisting of all maps $x \in \mathbb{R}^{\binom{V}{2}}$ with

$$0 \leq x(u, v) \leq 1 \tag{3.1}$$

for any two distinct elements $u, v \in V$,

- the convex polytope $P = P(V) \subset [0, 1]^{\binom{V}{2}}$ consisting of all vectors $x$ in $[0, 1]^{\binom{V}{2}}$ for which, in addition, the inequality

$$x(u, v) + x(v, w) - x(w, u) \leq 1 \tag{3.2}$$

holds for any three distinct elements $u, v, w \in V$,

- and the set $P_0$ consisting of all vertices in $P$.

We note that

- the set of vertices of $[0, 1]^{\binom{V}{2}}$ coincides with the set $\{0, 1\}^{\binom{V}{2}}$ of all maps from $\binom{V}{2}$ into the 2-subset $\{0, 1\}$ of $[0, 1]$,
- $P \cap \{0, 1\}^{\binom{V}{2}} \subseteq P_0$ holds,
- and there is a canonical one-to-one correspondence between
  - (i) those vertices $x \in \{0, 1\}^{\binom{V}{2}}$ of $[0, 1]^{\binom{V}{2}}$ that are contained in $P$ (and, hence, in $P_0$)
  - (ii) and community networks or, equivalently, partitions $\Pi$ of $V$ into a disjoint union of subsets.

Indeed, associating to each vertex $x \in \{0, 1\}^{\binom{V}{2}}$ of $[0, 1]^{\binom{V}{2}}$, the graph

$$G_x := (V, E_x := \{\{u, v\} \in \binom{V}{2} : x(u, v) = 1\})$$

(where "$x := y$" means "we define the term $x$ by requiring it to mean $y$"), it has been noted by Grötschel and Wakabayashi [15, 16] that $G_x$ is a community network if and only if $x$ satisfies the *constraints* defined by (3.1) and (3.2). Thus, the integral-valued maps $x \in \mathbb{R}^{\binom{V}{2}}$ that satisfy these constraints parameterize, in a one-to-one fashion, the community networks that we want to investigate and among which we want to identify, for any given finite simple graph $G$, that one which approximates $G$ best.

To this end, we proposed in [7] to associate, to any given finite simple graph $G = (V, E)$ with vertex set $V$ and edge set $E \subseteq \binom{V}{2}$, the one-parameter family $LP_G(s)$ of linear-programming problems of finding, for every $s \geq 1$, those maps

$$x = x(G, s) \in P \subset \mathbb{R}^{\binom{V}{2}}$$

that satisfy (i) the constraints defined by (3.1) and (3.2) and (ii) optimize the linear form

$$\ell_G^s : \mathbb{R}^{\binom{V}{2}} \to \mathbb{R} : x \mapsto s \sum_{\{u,v\} \in E} x(u, v) - \sum_{\{u,v\} \in \binom{V}{2} - E} x(u, v) \ \ (s \geq 1) \tag{3.3}$$

defined on $\mathbb{R}^{\binom{V}{2}}$. We used the software CPLEX 9.1 to solve these linear-programming problems for various input graphs[2].

We observed that, as noted already in the introduction,

- there exists one positive real number $\bar{s} = s(G) \geq 1$ such that, for every $s \geq \bar{s}$, there is only one vertex $x$ in $P_0$ — and, therefore, only one map $x$ in $P$ — that maximizes the linear form $\ell_G^s$,
- all values $x(u, v)$ are necessarily either 0 or 1,
- the corresponding partition $\pi_0(G_x)$ of $V$ into the connected components of $G_x$ coincides with the partition $\pi_0(G)$ of $V$ into the set of connected components of $G$, i.e., $x$ coincides with the map $x^G$ defined by $x^G(u, v) := 0$ if $u$ and $v$ are contained in distinct components of $G$, and $x^G(u, v) := 1$ else.

We observed also that, much to our own surprise, denoting by $s^*(G)$ the smallest value $\geq 1$ of our control parameter $s$ for which the LP problem $LP_G(s)$ has an integer-valued solution, the associated community network $G' := G_{x(G,s^*(G))}$ almost invariably coincides with a community network considered to provide one of, if not "the best" approximation of $G$.

Based on this finding, we explored in [7] the following simple strategy for detecting community structures associated to a given graph $G$:

- Start with $s := 1$.
- Use CPLEX 9.1 (or any other good software tool for solving LP problems) to find vertices in $P_0$ that solve the linear programming problem $LP_G(s)$.
- Increase $s$ continuously in sufficiently small steps until the smallest value $s^* = s^*(G) \in [1, +\infty)$ for which this problem has an integer solution $x^*(G) := x(G, s^*(G))$ is found[3].

[2]Actually, we also studied other variants of one-parameter families in [7], but will restrict our attention here to this particularly simple choice.

[3]A more direct and systematic "geometric" approach to finding exactly all values of $s$ at which $x(G, s)$ is about to change is presently under development.

- Then stop and consider the partition $\Pi(G) := \Pi_{x^*(G)}$ as a hopefully reasonably good solution of the original problem, i.e., the problem of finding a "good" community structure approximating the input graph $G$ (provided there exists a good approximation for $G$ at all).

We demonstrated that this strategy yields indeed pretty good solutions for some well known benchmark problems including *Zachary's Karate Club* [39] and the *Chesapeake-Bay Food Web* compiled by Baird and Ulanowicz [3]: For Zachary's Karate club, our method produced exactly the "historically correct" partition. Regarding the Chesapeake-Bay food web, our result was checked by Robert Ulanowicz who, comparing our result with that of other algorithms, judged that "both groupings are quite good, but — by placing *blue crab* correctly among the *benthic feeders* — you win the competition probably by a hair".

# 4 The Perturbation Experiments

Next, we will investigate the reliability of the LP-based approach by analyzing a number of more and more randomized test cases. To begin with, we start with a given "target partition" $\Pi$ of $V$ and consider the associated community network $H_\Pi = (V, E_\Pi)$ whose vertex set is $V$ while its edge set $E_\Pi$ coincides with $\bigcup_{U\in\Pi} \binom{U}{2}$.

Clearly, a graph $G' = (V, E')$ is a community network if and only if it is of the form $G' = H_{\Pi'}$ for some partition $\Pi'$ of $V$ in which case the partition $\pi_0(G')$ of the vertex set $V$ into the connected components of $G'$ is the unique partition $\Pi'$ of $V$ with $G' = H_{\Pi'}$.

Next, we

(i) randomly generate (Erdös-Renyi) graphs $R := (V, F)$ with the same vertex set $V$ and more and more edges,

(ii) form the *symmetric difference* $F_\Pi := E_\Pi \Delta F$,

(iii) apply our algorithm to $H_\Pi \Delta R := (V, F_\Pi)$,

(iv) define the *perturbation ratio* $p(\Pi, R)$ of $R$ relative to $\Pi$ to be the quotient $\frac{|F|}{|E_\Pi|}$ of the cardinality $|F|$ of $F$ by that of $E_\Pi$,

(v) and expect that, at least for small perturbation ratios, the resulting partitions $\Pi(H_\Pi \Delta R)$ of $V$ should not differ too much from — or even coincide with — the target partition $\Pi$.

To check, more specifically, for which perturbation ratios this expectation is justified, we generated ten times ten random graphs $R_i^j$ $(i, j = 1, 2, \ldots, 10)$, ten for each perturbation ratio $p_i \approx \frac{i}{10}$ $(i = 1, 2, \ldots, 10)$, and compared the resulting partitions $\Pi_i^j := \Pi(H_\Pi \Delta R_i^j)$ $(i, j = 1, 2, \ldots, 10)$ with the original partition $\Pi$.

We applied this procedure to a partition $\Pi = \Pi_{12|9|8|6}$ consisting of four disjoint sets of cardinality $12, 9, 8$, and $6$ (for which $E_\Pi$ consists of altogether $66+36+28+15 = 145$ edges), and correspondingly defined partitions $\Pi_{15|10|10}$ and $\Pi_{16|8|8|8}$.

In Figure 1 and Figure2, we present four sample graphs (in a way that, by keeping the vertices in each clique close together at their "original" positions so that it should be easy to recognize the original partition as long as this is possible at all) that we obtained from $\Pi_{12|9|8|6}$ for the perturbation ratios $0.4, 0.8, 0.85$ and $0.9$. Clearly, when the perturbation ratio is as low as $0.4$, the community structure is detected easily. In case $p_i \approx 0.8$, the original structure gets blurry, but can, in most cases, still be guessed correctly. When $p_i \approx 0.9$, the original structure becomes essentially irrecognizable — in-spite of the specific presentation derived from the input partition.

We now present the results of the experiments. Referring first to the four sample graphs depicted in Figure 1 and Figure 2, the original partition could be detected, as expected, almost exactly with our algorithm in case $p_i \leq 0.85$. In case $p_i \approx 0.9$, our algorithm returns a partition with one subset that is the union of the two original subsets of size 12 and 8 while the other two subsets in $\Pi$ are still detected exactly.

To systematically compare the partitions $\Pi_i^j$ with the target partition, we define

(i) the *transfer distance* $D(\Pi_1, \Pi_2) = D_{transfer}(\Pi_1, \Pi_2)$ between any two partitions $\Pi_1, \Pi_2$ to be the minimal number of "single vertex moves" (moving one single vertices from one subset to another one at a time) that are required to change $\Pi_1$ into $\Pi_2$ (cf. [6]),

(ii) the *maintenance ratio* by

$$M(\Pi_1, \Pi_2) = M_{transfer}(\Pi_1, \Pi_2) := \big(n - D(\Pi_1, \Pi_2)\big)/n,$$

where $n$ is the number of vertices in the network,

(iii) and the *average maintenance ratio* by

$$AM(\Pi, i) := \frac{1}{10} \sum_{j=1}^{10} M(\Pi, \Pi_i^j) \ \ (i = 1, 2, \ldots, 10),$$

and plot the perturbation ratio versus the average maintenance ratio (cf. Figure 3) — including also, as the maintenance ratio declines rapidly just above $p \approx 0.8$ for all three test systems, the ratio $p \approx 0.85$.

We also applied the CNM algorithm (as available from the internet, cf. [17]) to our 3 sets of examples. The results are presented in Figure 3: The solid lines represent the results obtained by the LP-based method, and the dash-dot lines represent those obtained by the CNM algorithm — the "star", the "dot", and the "triangle" designating the three test systems, respectively. Much to our own surprise, the LP-based method performed consistently better than the

modularity-based algorithm for all perturbation ratios below 1. Only when the perturbation ratio approaches 1 and the number of perturbed edges approaches the total number of edges in the original community network, the two methods perform about equally poor.

We wondered also whether comparing the two methods using the above definition of maintenance ratios might, for one reason or the other, not be a fair deal with respect to the CNM algorithm and whether, e.g., allowing to move whole subsets rather than single elements in one go might be "fairer" because there should be a difference between, say, the distance of the partition

$$\Pi_1 := \{\{1,2,3,4\},\{5,6,7,8\},\{9,10,11,12\}\}$$

to the partition

$$\Pi_2 := \{\{1,2,3,4,5,6,7,8\},\{9,10,11,12\}\},$$

and that to the partition

$$\Pi_2' := \{\{1,2,3,4,5,6\},\{7,8,9,10,11,12\}\}.$$

For this reason, we defined the *subset-transfer distance* $STD(\Pi_1,\Pi_2)$ as the minimal number of moves of *feasible subsets*, i.e., non-empty intersections $A_1 \cap A_2$ of the sets $A_1 \in \Pi_1$ and $A_2 \in \Pi_2$ in the two partitions $\Pi_1$ and $\Pi_2$ under consideration (rather than single elements) that are required to change $\Pi_1$ into $\Pi_2$, and denoted the corresponding subset-maintenance ratio by

$$STM(\Pi_1,\Pi_2) := (N - STD(\Pi_1,\Pi_2))/N,$$

where $N = N(\Pi_1,\Pi_2)$ is the total number of feasible subsets.

In Figure 4, we plot the average values of the subset-maintenance ratio $ASM$. Apparently, this did not help much and even for perturbation ratios above 0.8 where almost total randomization seems to set in, they remain notably larger for the LP-based method.

We also used another popular parameter to compare two partitions, the *adjusted Rand index* proposed by L. Hubert and P. Arabie (see [18, 30] for detailed definitions). This parameter is expected to be larger for more similar partitions, and it has a maximum value of 1 that is achieved when the two partitions coincide. We compare the results obtained with both methods in Figure 5: Again, the LP-based method seems to perform better.

We also examined the following question: If we choose just $k$ elements in $\binom{V}{2}$ randomly out of the altogether $\binom{35}{2} = 595$ edges in the complete graph with a vertex set of cardinality 35, we have to expect that, considering the partition $\Pi$ of $V$ into four subsets of cardinality $12, 9, 8$, and 6, respectively, approximately $k\,\frac{145}{595}$ of those will be contained in the edge set $E_\Pi$ of $H_\Pi$ and will, therefore, be removed while $k\,\frac{450}{595}$, that is, slightly more than three times as many, are contained in the complement of that edge set and will, therefore, be inserted. For the perturbation ratio 1, this means that about 35 edges will be removed and about 110 will be

inserted. So altogether, we have to expect that the edge set of the resulting graph will have approximately $k\,\frac{305}{595} \approx \frac{k}{2}$ more edges than $H_\Pi$. But, as in most "real-world" examples, we would expect to "observe" rather too few than too many edges, we considered the effect of the following procedure:

(i) Specify, for any given (total) perturbation ratio $p_i$ two numbers, the "deletion ratio" $del_i$ and the "insertion ratio" $ins_i$ for which we assume that $del_i + ins_i = p_i$ holds,

(ii) choose about $145\,del_i$ edges from the 145 original edges in $E_\Pi$,

(iii) and about $145\,ins_i$ edges from the remaining $595 - 145 = 450$ edges,

(iv) then delete those $145\,del_i$ edges and insert those $145\,ins_i$ edges we chose.

For the perturbation ratios $p_i = 0.8$ and $p_i = 0.9$, and different pairs $(del_i, ins_i)$, we generated four examples for each pair and tested the performance of the two algorithm. We compared the results using the average values of the maintenance ratio with respect to single-element transfer and subset-transfer and the adjusted Rand index (cf. Figures 6 - 8).

Generally speaking, for low deletion ratio, the LP-based method performs better than the CNM method. However, when the deletion ratios increase to around $del_i = 0.6$ and higher, the maintenance ratios decrease drastically for both methods while, remarkably, the CNM method consistently produces slightly better results. So, some information that the present form of the LP-based algorithm does not yet detect must still be there in that case, and it will be worthwhile to investigate how this algorithm can perhaps be improved to also detect this remaining bit of information.

Finally, we note that comparisons based on subset transfer are not always consistent with those using single-element transfer.

# 5 The "Four-Groups" Experiments

We continue investigating the reliability of the LP-based approach by analyzing how it performs when applied to artificial networks that are constructed as follows (cf.[14]): Each network contains 128 vertices, divided into four groups of 32 vertices each. All vertices have fixed average degree $k = 16$, they are connected randomly to the members of the same community by an average of $k_{in}$ edges, and to members of the other communities by an average of $k_{out} := 16 - k_{in}$ edges. This design produces networks with "known" community structure. However, as the value of $k_{out}$ increases, it will become more and more difficult to detect it.

These "four-groups" experiments are much-studied benchmark experiments and were performed in [9, 14, 25, 26, 29, 31]. The resulting maintenance ratios based on single-element transfer are plotted, as a function of $k_{out}$, in Figure 9.

Comparing our results with those obtained by the CNM algorithm, our approach performs about just as well. More specifically, the LP-based method does not seem to produce a single mistake for $k_{out} < 6$ while the associated maintenance ratio decreases gradually only for $k_{out} > 6$ in which range the CNM algorithm (and some other methods) appears to perform slightly better. Thus, once again, one gets tempted to search for a *combination therapy* that incorporates the best ideas of all methods and may outperform all of them. E.g., recalling the *betweenness* parameter from [14], one may try to penalize insertion of (not yet existing) edges $\{u, v\}$ in proportion to the distance of $u$ and $v$ (in the original network).

# 6 Another "Real-Life" Example

To conclude this paper, we present yet another "real-life" example: We analyze a network containing 101 proteins studied by A. Pocklington et al. [27] and compare our result with that obtained in [27] using the algorithm described in [26]. While this algorithm detected 13 communities all of which appear to be associated to a specific function, we only obtain six. There are 4 communities detected by both methods, the other two new communities are unions of old communities (see Figure 10: The rectangles indicate the old communities labeled by numbers, and the ellipses indicate the new communities labeled by letters).

We investigated also the sub-community structures for the two largest communities $E$ and $F$ using the LP-based algorithm. For Community $E$, three communities were detected. The first one coincides with the old community 8, and the union of the second and the third one coincides with the old community 2, the second one being formed by three proteins belonging to the group of *synaptic vesicles*.

Within the Community $F$, we identified three sub-communities. The first one is the union of the communities 6 and 7, the second one is the union of the communities 1, 10, 13 and some proteins from the community 3, and the third one is the union of the community 9 and the remaining proteins in 3. All the sub-communities inherit the specific functions from the larger ones, suggesting that our approach yields some new information regarding the relationship between the involved proteins not obtained by the algorithm used by Pocklington et al..

# 7 Final Remarks

It is an obvious advantage of our approach that — just as the CNM method — it does provide a "natural" way to directly detect a network's community structure. In contrast, many other methods developed so far for community detection require users to do *prune* the resulting system of potential communities to identify a proper community structure in a way controlled by some often not very transparent parameters.

In addition, our approach has a flexibility allowing the user to incorporate and test any additional information he might deem useful. E.g., if one wants to study only partitions that split the given set $V$ of agents into at most two parts, only, all one has to do is to add, for any three distinct $u, v, w \in V$, the inequality

$$x(u,v) + x(v,w) + x(w,u) \geq 1 \tag{7.4}$$

to our list of constraints. And one can, of course, also play with the penalty function to check all sorts of variants of the algorithm.

Regarding the speed of our algorithm, one should note that polynomial algorithms exist for LP problems only "in theory" while the potentially exponential simplex method performs great in most cases — actually, it is provably almost linear "in average". We believe that, without substantial improvement, the current form of our algorithm cannot deal with more than, at most, a few hundreds of vertices, but would hope that it can become much faster using software tools dedicated to exactly dealing with the specific LP tasks we have been dealing with here.

**Acknowledgments.** The authors are grateful to the reviewers for their constructive criticism, to A. Clauset for allowing us to use his algorithm proposed in his joint work with Newman and Moore, to J. Reichardt for helping us to generate the data on the four-groups experiment, and to Q. H. Hou, Roger Q. L. Yu, and X. Y. Zhang for helpful conversations. This work was supported by the 973 Project on Mathematical Mechanization, the PCSIRT Project of the Ministry of Education, the Ministry of Science and Technology, the National Science Foundation of China, and the Max Planck Society.

# References

[1] Albert, R., Jeong, H., and Barabási, A.-L.: 'Internet: Diameter of the World-Wide Web', *Nature*, 1999, 401, pp. 130-131

[2] Amaral, L. A. N., Scala, A., Barthélémy, M., and Stanley, H. E.: 'Classes of small-world networks', *Proc. Natl. Acad. Sci. USA*, 2000, 97, pp. 11149-11152

[3] Baird, D., and Ulanowicz, R. E.: 'The seasonal dynamics of the Chesapeake Bay ecosystem', *Ecological Monographs*, 1989, 59, (4), pp. 329-364

[4] Bansal, N., Blum, A., and Chawla, S.: 'Correlation clustering', *Mach. Learn.*, 2004, 56, pp. 89-113

[5] Barabási, A.-L., and Albert, R.: 'Emergence of scaling in random networks', *Science*, 1999, 286, pp. 509-512

[6] Charon, I., Denoeud, L., Guénoche, A., and Hudry, O.: 'Maximum transfer distance between partitions', *J. Classif.*, 2006, 23, (1), pp. 103-121

[7] Chen, W. Y. C., Dress, A. W. M., and Yu, W. Q.: 'Community structures of networks'. MACIS, Beijing, China, July 2006

[8] Clauset, A., Newman, M. E. J., and Moore, C.: 'Finding community structure in very large networks', *Phys. Rev. E*, 2004, 69, 026113

[9] Clauset, A.: 'Finding local community structure in networks', *Phys. Rev. E*, 2005, 72, 026132

[10] Davidson, E., et al.: 'A genomic regulatory network for development', *Science*, 2002, 295, pp. 1669-1678

[11] Demaine, E., and Immorlica, N.: 'Correlation clustering with partial information', *LNCS*, 2003, 2764, pp. 1-13

[12] Flake, G. W., Lawrence, S. R., Giles, C. L., and Coetzee, F. M.: 'Self-organization and identification of Web communities', *IEEE Computer*, 2002, 35, pp. 66-71

[13] Frank, K.: 'Identifying cohesive subgroups', *Soc. Networks*, 1995, 17, pp. 27-56

[14] Girvan, M., and Newman, M. E. J.: 'Community structure in social and biological networks', *Proc. Natl. Acad. Sci. USA*, 2002, 99, pp. 7821-7826

[15] Grötschel, M., and Wakabayashi, Y.: 'A cutting plane algorithm for a clustering problem', *Math. Programming*, 1989, 45, pp. 59-96

[16] Grötschel, M., and Wakabayashi, Y.: 'Facets of the subset partitioning polytope', *Math. Programming*, 1990, 47, pp. 367-387

[17] http://www.cs.unm.edu/~aaron/research/fastmodularity.htm, accessed August 2006

[18] Hubert, L., and Arabie, P.: 'Comparing partitions', *J. Classif.*, 1985, 2, pp. 193-218

[19] Jeong, H., Mason, S. P., Barabási, A.-L., and Oltvai, Z. N.: 'Lethality and centrality in protein networks', *Nature*, 2001, 411, pp. 41-42

[20] Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N., and Oltvai, A.-L.: 'The large-scale organization of metabolic networks', *Nature*, 2000, 407, pp. 651-654

[21] Kernighan, B. W., and Lin, S.: 'An efficient heuristic procedure for partitioning graphs', *Bell System Technical Journal*, 1970, 49, pp. 291-307

[22] Kleinberg, J., and Lawrence, S.: 'The structure of the Web', *Science*, 2001, 294, pp. 1849-1850

[23] Krause, A. E., Frank, K. A., Mason, D. M., Ulanowicz, R. E., and Taylor, W. W.: 'Compartments revealed in food-web structure', *Nature*, 2003, 426, pp. 282-285

[24] Newman, M. E. J.: 'The structure of scientific collaboration networks', *Proc. Natl. Acad. Sci. USA*, 2001, 98, pp. 404-409

[25] Newman, M. E. J.: 'Fast algorithms for detecting community structure in networks', *Phys. Rev. E*, 2004, 69, 066133

[26] Newman, M. E. J., and Girvan, M.: 'Finding and evaluating community structure in networks', *Phys. Rev. E*, 2004, 69, 026113

[27] Pocklington, A., Cumiskey, M., Armstrong, J., and Grant, S.: 'The proteomes of neurotransmitter receptor complexes form modular networks with distributed functionality underlying plasticity and behaviour', *Mol. Syst. Biol.*, 2006, 2, 2006.0023

[28] Pothen, A., Simon, H., and Liou, K.-P.: 'Partitioning sparse matrices with eigenvectors of graphs', *SIAM J. Matrix Anal. Appl.* , 1990, 11, pp. 430-452

[29] Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., and Parisi, D.: 'Defining and identifying communities in networks', *Proc. Natl. Acad. Sci. USA*, 2004, 101, pp. 2658-2663

[30] Rand, W. M.: 'Objective criteria for the evaluation of clustering methods', *J. Am. Stat. Assoc.*, 1971, 66, pp. 846-850

[31] Reichardt, J., and Bornholdt, S.: 'Detecting fuzzy community structures in complex networks with a Potts model', *Phys. Rev. Lett.*, 2004, 93, 218701

[32] Strogatz, S. H.: 'Exploring complex networks', *Nature*, 2001, 410, pp. 268-276

[33] Tyler, J. R., Wilkinson, D. M., and Huberman, B. A.: 'Email as spectroscopy: Automated discovery of community structure within organizations'. Proc. 1st Int. Conf. Communities and Technologies, Amsterdam, Holland, Sept. 2003, pp. 81-96

[34] Ulanowicz, R. E., and Wolff, W. F.: 'Ecosystem flow networks: Loaded dice?', *Math. Biosci.*, 1991, 103, pp. 45-68

[35] Valente A., and Cusick, M.: 'Yeast protein interactome topology provides framework for coordinated-functionality', *Nucleic Acids Res.*, 2006, 34, pp. 2812-2819

[36] Wasserman S., and Faust, K.: 'Social network analysis' (Cambridge University Press, Cambridge, UK, 1994)

[37] Watts, D. J., and Strogatz, S. H.: 'Collective dynamics of 'small world' networks', *Nature*, 1998, 393, pp. 440-442

[38] Wu, F., and Huberman, B. A.: 'Finding communities in linear time: A physics approach', *Eur. Phys. J. B*, 2004, 38, pp. 331-338

[39] Zachary, W. W.: 'An information flow model for conflict and fission in small groups', *J. Anthropological Research*, 1977, 33, pp. 452-473

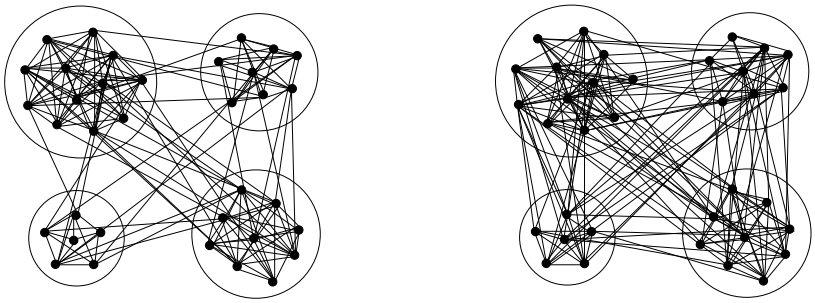

Figure 1: Two samples with $p_i \approx 0.4$ (left) and $p_i \approx 0.8$, respectively. Circles were drawn for highlighting the "original" four groups.

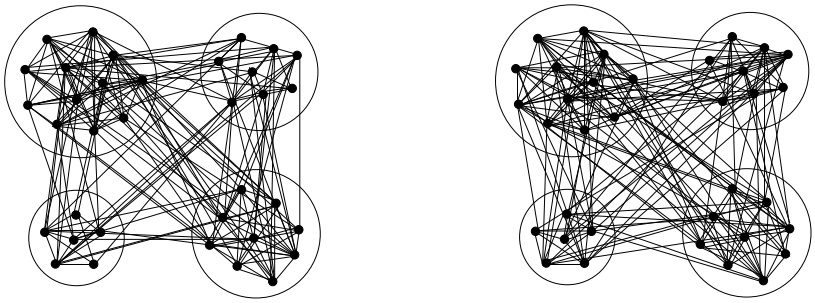

Figure 2: Two samples with $p_i \approx 0.85$ (left) and $p_i \approx 0.9$, respectively. Circles were drawn for highlighting the "original" four groups.

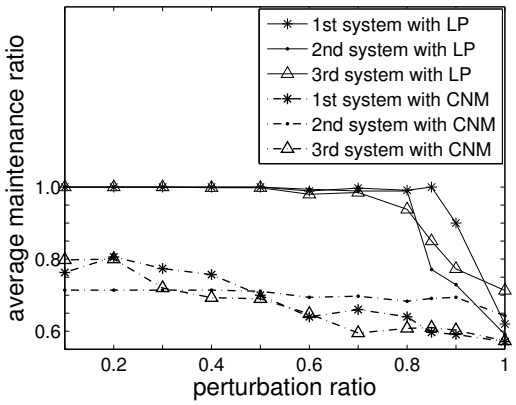


Figure 3: Comparison of the single-element transfer distance for the two methods.

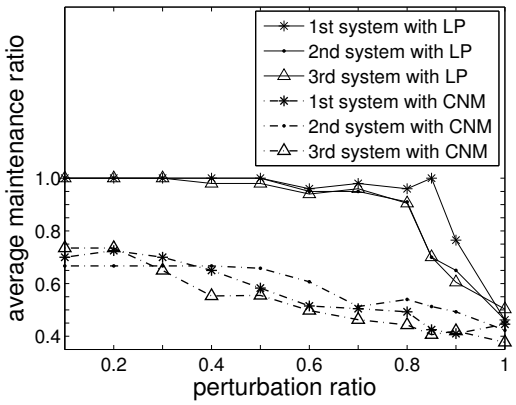


Figure 4: Comparison of the subset-transfer distance for the two methods.

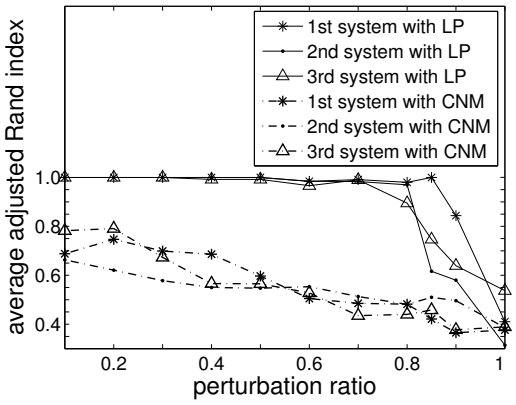


Figure 5: Comparison of the adjusted Rand index for the two methods.

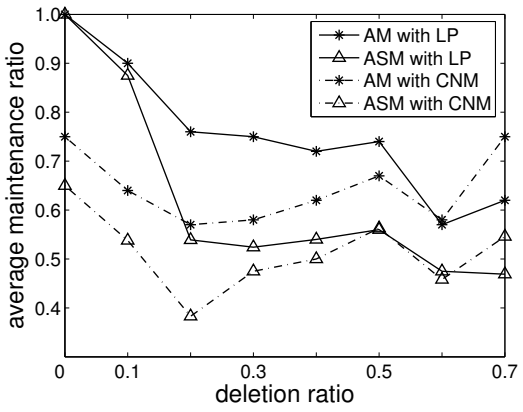


Figure 6: Comparison of the maintenance ratios for the two methods for $p_i = 0.8$.

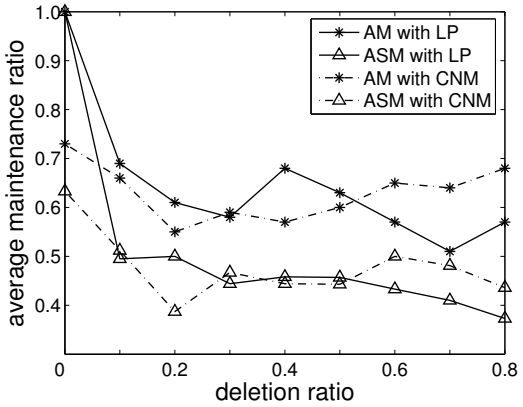


Figure 7: Comparison of the maintenance ratios for the two methods for $p_i = 0.9$.

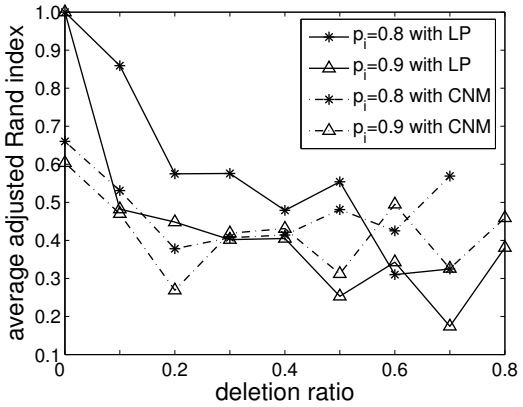


Figure 8: Comparison of the adjusted Rand index for two methods for $p_i = 0.8$ and $p_i = 0.9$.

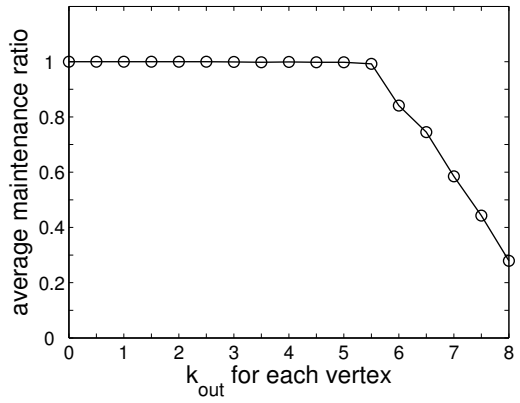


Figure 9: The community result for the four-groups experiments.

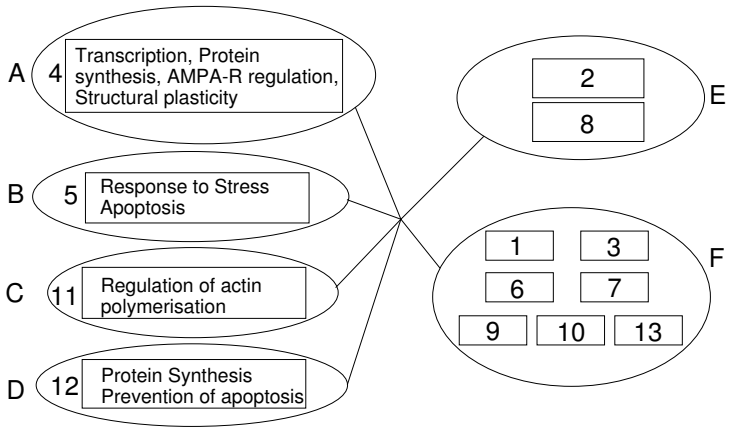


Figure 10: The community result for the 101 proteins network.