# Robust Optimization of Graph Partitioning and Critical Node Detection in Analyzing Networks

Neng Fan and Panos M. Pardalos

Center for Applied Optimization, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL, USA
{andynfan,pardalos}@ufl.edu

**Abstract.** The graph partitioning problem (GPP) consists of partitioning the vertex set of a graph into several disjoint subsets so that the sum of weights of the edges between the disjoint subsets is minimized. The critical node problem (CNP) is to detect a set of vertices in a graph whose deletion results in the graph having the minimum pairwise connectivity between the remaining vertices. Both GPP and CNP find many applications in identification of community structures or influential individuals in social networks, telecommunication networks, and supply chain networks. In this paper, we use integer programming to formulate GPP and CNP. In several practice cases, we have networks with uncertain weights of links. Some times, these uncertainties have no information of probability distribution. We use robust optimization models of GPP and CNP to formulate the community structures or influential individuals in such networks.

## 1 Introduction

The graph partitioning problem (GPP) consists of partitioning the vertex set of a graph into several disjoint subsets so that the sum of weights of the edges between the disjoint subsets is minimized. The critical node problem (CNP) is to detect a set of vertices in a graph whose deletion results in the graph having the minimum pairwise connectivity between the remaining vertices. Both GPP and CNP are NP-complete [8,1].

Let $G = (V, E)$ be an undirected graph with a set of vertices $V = \{v_1, v_2, \cdots, v_N\}$ and a set of edges $E = \{(v_i, v_j) :$ edge between vertices $v_i$ and $v_j, 1 \le i, j \le N\}$, where $N$ is the number of vertices. The weights of the edges are given by a matrix $W = (w_{ij})_{N\times N}$, where $w_{ij}(> 0)$ denotes the weight of edge $(v_i, v_j)$ and $w_{ij} = 0$ if no edge $(v_i, v_j)$ exists between vertices $v_i$ and $v_j$. This matrix is symmetric for undirected graphs $G$ and is the adjacency matrix of $G$ if $w_{ij} \in \{0, 1\}$.

For the graph partitioning problem, we are given the cardinalities $n_1, \cdots, n_K$ of subsets that we want to partition $V$, and $K$ is the number of subsets. Let $x_{ik}$ be the indicator that vertex $v_i$ belongs to the $k$th subset if $x_{ik} = 1$ or not if $x_{ik} = 0$, and $y_{ij}$ be the indicator that the edge $(v_i, v_j)$ with vertices $v_i, v_j$ are in different subsets if $y_{ij} = 1$ and $v_i, v_j$ in the same subset if $y_{ij} = 0$. Thus, the sum of weights of the edges between the disjoint subsets can be expressed as $\frac{1}{2}\sum_{i=1}^N\sum_{j=1}^N w_{ij}y_{ij}$ or $\sum_{i=1}^N\sum_{j=i+1}^N w_{ij}y_{ij}$ because of $w_{ij} = w_{ji}$ and $w_{ii} = 0$ for non-existence of loops. Each vertex $v_i$ has to be partitioned into one and only one subset, i.e., $\sum_{k=1}^K x_{ik} = 1$, and the $k$th subset has the number $n_k$ of vertices, i.e., $\sum_{i=1}^N x_{ik} = n_k$. The relation between $x_{ik}$ and $y_{ij}$ can be expressed as $y_{ij} = 1 - \sum_{k=1}^K x_{ik}x_{jk}$

W. Wu and O. Daescu (Eds.): COCOA 2010, Part I, LNCS 6508, pp. 170–183, 2010.

and this can be linearized as $-y_{ij} - x_{ik} + x_{jk} \leq 0, -y_{ij} + x_{ik} - x_{jk} \leq 0$ for $k = 1, \cdots, K$ under the objective of minimization. Therefore, the feasible set of deterministic formulation of graph partitioning problem for a graph $G = (V, E)$ with weight matrix $W$ is

$$X = \left\{ (x_{ik}, y_{ij}) : \begin{array}{l} \sum_{k=1}^{K} x_{ik} = 1, \sum_{i=1}^{N} x_{ik} = n_k, \\ -y_{ij} - x_{ik} + x_{jk} \leq 0, \\ -y_{ij} + x_{ik} - x_{jk} \leq 0, \\ x_{ik} \in \{0,1\}, y_{ij} \in \{0,1\}, \\ \quad i = 1, \cdots, N, j = i+1, \cdots, N, k = 1, \cdots, K \end{array} \right\}, \tag{1}$$

and the objective function is

$$\min_{(x_{ik}, y_{ij}) \in X} \sum_{i=1}^{N} \sum_{j=i+1}^{N} w_{ij} y_{ij} \tag{2}$$

where we minimize the total weight of edges connecting distinct subsets. The GPP is to solve the program with the objective (2) and the constraints in (1) of $X$. This is a binary integer linear programming problem.

For the critical node problem, we are given the number $K$ as the number of vertices we want to delete in $V$. Let $v_i$ be the indicator that vertex $v_i$ belongs to the deleted subset $V_d$ of $V$ if $v_i = 1$ and otherwise $v_i = 0$, and let $u_{ij}$ be the indicator that the edge $(v_i, v_j)$ with two ends $v_i, v_j$ are in the resulted graph after deletion of the subset $V_d$ of $V$ if $u_{ij} = 1$ and otherwise $u_{ij} = 0$. Here, we use the notation $v_i$ as one vertex of $V$ and the indicator of this vertex to be deleted or not. Thus, the pairwise connectivity between the remaining vertices in $V \setminus V_d$ can be expressed as $\sum_{i=1}^{N} \sum_{j=i+1}^{N} w_{ij} u_{ij}$ because of the symmetric $w_{ij} = w_{ji}$ and no loop of $G$ with $w_{ii} = 0$. The constraints of CNP include that the number of vertices in the deleted subset is $K$, i.e., $\sum_{i=1}^{N} v_i = K$, and all three edges in $E$ have the relation that if two edges are in the resulted graph, another edge is also in the resulted graph, i.e., $\max\{u_{ij} + u_{jk} - u_{ik}, u_{ij} - u_{jk} + u_{ik}, -u_{ij} + u_{jk} + u_{ik}\} \leq 1$. The relation between $u_{ij}$ and $v_i, v_j$ can be expressed as $u_{ij} + v_i + v_j \geq 1$ under the objective of minimization, which implies that the edge $(v_i, v_j)$ between the subsets $V_d$ and $V \setminus V_d$ should be deleted if one or two of the vertices are deleted. Therefore, the feasible set of deterministic formulation of critical node problem for a graph $G = (V, E)$ with weight matrix $W$ is

$$Y = \left\{ (v_i, u_{ij}) : \begin{array}{l} u_{ij} + v_i + v_j \geq 1, \\ u_{ij} + u_{jk} - u_{ik} \leq 1, \\ u_{ij} - u_{jk} + u_{ik} \leq 1, \\ -u_{ij} + u_{jk} + u_{ik} \leq 1, \\ \sum_{i=1}^{N} v_i = K, \\ v_i \in \{0,1\}, u_{ij} \in \{0,1\}, \\ \quad i = 1, \cdots, N, j = i+1, \cdots, N, k = j+1, \cdots, N \end{array} \right\}, \tag{3}$$

and the objective function is

$$\min_{(v_i, u_{ij}) \in Y} \sum_{i=1}^{N} \sum_{j=i+1}^{N} w_{ij} u_{ij} \tag{4}$$

where we minimize the pairwise connectivity between the remaining vertices. The CNP is to solve the program with the objective (4) and the constraints in (3) of $Y$. This is also a binary integer linear programming problem.

The graph partitioning problem has been studied for a long time and recently studied by linear and quadratic programming approaches [6]. The critical node problem is proposed recently by [1], and solved with several heuristic methods and exact integer programming methods. In this paper, we minimize the pairwise connectivity between the remaining vertices after deleting some vertices instead of another form named the cardinality constrained critical node problem, which is to minimize the number of deleted nodes to limit the connectivity.

Both GPP and CNP have been applied in analyzing networks [5,2,4]. Two important elements or structures in networks are communities and influential individuals, where a community is a dense group of nodes with high connectivity and the influential individual is a node which plays a leader role in the network. The GPP is to partition the nodes into several subsets which are quite related to communities, while the CNP is trying to find the critical nodes or influential individuals in the network. However, the influential individual or the critical node always have the dense part with itself as the center of this community. This leads us to use both GPP and CNP to analyze the networks in the meantime.

In practice, the links between nodes always change their connectivity in a network. That is, the weights are uncertain along the time. In this paper, we use robust optimization models to formulate both GPP and CNP to deal with these problems with uncertain weights. In addition, we use our models to analyze several networks arising from social networks or some artificial networks.

The rest of this paper is organized as follows: section 2 discusses the robust optimization models and algorithms based on a decomposition method, for GPP and CNP with uncertain weights; In section 3, we discuss the application of GPP and CNP in networks with detailed explanations; In section 4, several numerical experiments are performed to analyze some social networks; Section 5 concludes the paper.

# 2 Robust Models for GPP and CNP

## 2.1 Uncertainty Assumptions

In this paper, we consider the uncertainty for the weight matrix $W = (w_{ij})_{N\times N}$. Assume that each entry $w_{ij}$ is modeled as independent, symmetric and bounded random but unknown distribution variable $\tilde{w}_{ij}$, with values in $[w_{ij} - \hat{w}_{ij}, w_{ij} + \hat{w}_{ij}]$. Note that we require $w_{ij} = w_{ji}$ for undirected graph $G$ and thus $\hat{w}_{ij} = \hat{w}_{ji}$ for $i, j = 1,\cdots,N$. Assume $\hat{w}_{ij} \geq 0$, $w_{ij} \geq \hat{w}_{ij}$ and $w_{ii}, \hat{w}_{ii} = 0$ for all $i, j = 1,\cdots,N$.

## 2.2 Robust Optimization Models for GPP and CNP

For the graph $G = (V,E)$ with the weighted matrix $W = (w_{ij})_{N\times N}$, the uncertainties satisfy $\tilde{w}_{ij} \in [w_{ij} - \hat{w}_{ij}, w_{ij} + \hat{w}_{ij}]$. For the positive integer $K$, the GPP requires the given cardinalities $n_k (k = 1, \cdots, K)$. For general graph partitioning [6], $n_k$ is not necessarily to be given and only required to satisfy $n_k > 1$; For equal partitioning, $n_k \in \{\lfloor N/K \rfloor, \lfloor N/K \rfloor + 1\}$ with total $\sum_k n_k = N$. In CNP, we also delete exactly $K$ vertices to minimize the pairwise connectivity between the remaining vertices.

Because of the existence of uncertain weights of edges in $G$, we use robust optimization models to formulate the GPP and CNP in order to optimize against the worst cases by min-max objective functions. These models find the best partitioning in GPP and best deletion of CNP in the worst cases of the uncertainties $\tilde{w}_{ij}$.

Let $J$ be the index set of $W$ with uncertain changes, i.e., $J = \{(i, j) : \hat{w}_{ij} > 0, i = 1, \cdots, N, j = i+1, \cdots, N\}$, where we assume that $j > i$ since $W$ is symmetric. Let $\Gamma$ be a parameter, not necessarily integer, that takes values in the interval $[0, |J|]$. This parameter $\Gamma$, which is introduced in [3], is to adjust the robustness of the proposed method against the level of conservatism of the solution. The number of coefficients $w_{ij}$ is allowed to change up to $\lfloor \Gamma \rfloor$ and another $w_{i_t, j_t}$ changes by $(\Gamma - \lfloor \Gamma \rfloor)$.

Thus, the robust optimization model of GPP (RGPP) with given cardinalities $n_k$ can be formulated as follows,

$$\begin{aligned} \min_{(x_{ik}, y_{ij}) \in X} \quad & z \\ s.t. \quad & \sum_{i=1}^{N} \sum_{j=i+1}^{N} w_{ij} y_{ij} + \max_{\left\{\begin{array}{l} S : S \subseteq J, |S| \leq \Gamma \\ (i_t, j_t) \in J \setminus S \end{array}\right\}} \left( \sum_{(i,j) \in S} \hat{w}_{ij} y_{ij} + (\Gamma - \lfloor \Gamma \rfloor) \hat{w}_{i_t, j_t} y_{i_t, j_t} \right) - z \leq 0. \end{aligned} \tag{5}$$

and as shown in the following theorem, it can be reformulated as an equivalent binary integer linear programming. The method used in this proof was first proposed in [3].

**Theorem 1.** *The formulation* (5) *is equivalent to the following linear programming formulation:*

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} \sum_{j=i+1}^{N} w_{ij} y_{ij} + \Gamma p_0 + \sum_{(i,j) \in J} p_{ij} \\ s.t. \quad & p_0 + p_{ij} - \hat{w}_{ij} y_{ij} \geq 0, \quad (i, j) \in J \\ & p_{ij} \geq 0, \quad (i, j) \in J \\ & p_0 \geq 0, \\ & (x_{ik}, y_{ij}) \in X. \end{aligned} \tag{6}$$

*Proof.* For a given matrix $(y_{ij})_{i=1,\cdots,N, j=i+1,\cdots,N}$, the part

$$\max_{\left\{\begin{array}{l} S : S \subseteq J, |S| \leq \Gamma \\ (i_t, j_t) \in J \setminus S \end{array}\right\}} \left( \sum_{(i,j) \in S} \hat{w}_{ij} y_{ij} + (\Gamma - \lfloor \Gamma \rfloor) \hat{w}_{i_t, j_t} y_{i_t, j_t} \right),$$

in (5) can be linearized by introducing $z_{ij}$ for all $(i,j) \in J$ with the constraints $\sum_{(i,j)\in J} z_{ij} \leq \Gamma, 0 \leq z_{ij} \leq 1$, or equivalently, by the following formulation

$$\begin{aligned} \max \quad & \sum_{(i,j)\in J} \hat{w}_{ij} y_{ij} z_{ij} \\ s.t. \quad & \sum_{(i,j)\in J} z_{ij} \leq \Gamma, \\ & 0 \leq z_{ij} \leq 1, \quad (i,j) \in J \end{aligned} \tag{7}$$

The optimal solution of this formulation should have $\lfloor \Gamma \rfloor$ variables $z_{ij} = 1$ and one $z_{ij} = \Gamma - \lfloor \Gamma \rfloor$, which is equivalent to the optimal solution in the maximizing part in (5).

By strong duality, for a given matrix $(y_{ij})_{i=1,\cdots,N,j=i+1,\cdots,N}$, the problem (7) is linear and can be formulated as

$$\begin{aligned} \min \quad & \Gamma p_0 + \sum_{(i,j)\in J} p_{ij} \\ s.t. \quad & p_0 + p_{ij} - \hat{w}_{ij} y_{ij} \geq 0, \quad (i,j) \in J \\ & p_{ij} \geq 0, \quad (i,j) \in J \\ & p_0 \geq 0. \end{aligned}$$

Combing this formulation with (5), we obtain the equivalent formulation (6), which finishes the proof. □

Similarly, the robust optimization model for CNP (RCNP) to delete $K$ vertices is as follows,

$$\begin{aligned} \min_{(v_i,u_{ij})\in Y} \quad & z' \\ s.t. \quad & \sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{ij}u_{ij} + \max_{\left\{\begin{array}{l} S: S\subseteq J, |S|\leq \Gamma \\ (i_t,j_t)\in J\setminus S \end{array}\right\}} \left( \sum_{(i,j)\in S} \hat{w}_{ij}u_{ij} + (\Gamma - \lfloor \Gamma \rfloor)\hat{w}_{i_t,j_t}u_{i_t,j_t} \right) - z' \leq 0. \end{aligned} \tag{8}$$

and its equivalent binary integer linear programming formulation is

$$\begin{aligned} \min \quad & \sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{ij}u_{ij} + \Gamma p'_0 + \sum_{(i,j)\in J} p'_{ij} \\ s.t. \quad & p'_0 + p'_{ij} - \hat{w}_{ij}u_{ij} \geq 0, \quad (i,j) \in J, \\ & p'_{ij} \geq 0, \quad (i,j) \in J, \\ & p'_0 \geq 0, \\ & (v_i, u_{ij}) \in Y. \end{aligned} \tag{9}$$

By comparing the robust optimization models of GPP ((5),(6)) and CNP ((8),(9)), they are quite similar except that the feasible sets for $x_{ik}, y_{ij}$ in GPP and $v_i, u_{ij}$ in CNP. Both programs in (6) and (9) are formulated as binary integer linear programs, which can be solved through commercial software, such as CPLEX. In the following section, we use a decomposition method on a variable to reformulate robust optimization models of GPP and CNP. In addition, the probability distribution of the gap, between the objective value of the robust optimization model RGPP (5) and the objective value of the robust optimization model with respect to uncertainty $\tilde{w}_{ij}$ by choosing different $\Gamma$, is studied in [3]. Since the robust optimization model for RCNP (8) is quite similar to RGPP (5), the probability distribution of the gap for RCNP can be studied similarly.

### 2.3 Algorithm Based on a Decomposition on One Variable

Next, we will construct an algorithm based on a decomposition method on a variable to solve the programs (6) and (9). The numerical experiments in Section 4 show that this algorithm is much more efficient than the direct branch and bound method by CPLEX. For all $(i,j) \in J$, let $e_l$ ($l = 1,\cdots,|J|$) be the corresponding value of $\hat{w}_{ij}$ in the increasing order. For example, $e_1 = \min_{(i,j)\in J} \hat{w}_{ij}$ and $e_{|J|} = \max_{(i,j)\in J} \hat{w}_{ij}$. Let $(i^l, j^l) \in J$ be the corresponding index of $l$, i.e., $\hat{w}_{(i^l,j^l)} = e_l$. In addition, we define $e_0 = 0$. Thus, $[0,e_1],[e_1,e_2],\cdots,[e_{|J|},\infty)$ is a decomposition of $[0,\infty)$.

For $l = 0,1,\cdots,|J|$, we define the program $G^l$ as follows:

$$G^l = \Gamma e_l + \min_{(x_{ik},y_{ij})\in X} \left\{ \sum_{i=1}^{N} \sum_{j=i+1}^{N} w_{ij}y_{ij} + \sum_{(i,j):\hat{w}_{ij}\geq e_{l+1}} (\hat{w}_{ij} - e_l)y_{ij} \right\}. \tag{10}$$

Totally, there are $|J|+1$ of $G^l$s. In the following theorem, we prove that the decomposition method based on $p_0$ can solve the program (6). The method in the proof was first proposed in [3].

**Theorem 2.** *Solving robust graph partitioning problem* (6) *is equivalent to solve the* $|J|+1$ *problems* $G^l$*s in* (10) *for* $l = 0,1,\cdots,|J|$.

*Proof.* From (6), the optimal solution $(x^*_{ik}, y^*_{ij}, p^*_0, p^*_{ij})$ satisfies

$$p^*_{ij} = \max\{\hat{w}_{ij}y^*_{ij} - p^*_0, 0\},$$

and therefore, the objective function of (6) can be expressed as

$$\begin{aligned}
&\min_{\{p_0\geq 0,(x_{ik},y_{ij})\in X\}} \Gamma p_0 + \sum_{i=1}^{N} \sum_{j=i+1}^{N} w_{ij}y_{ij} + \sum_{(i,j)\in J} \max\{\hat{w}_{ij}y_{ij} - p_0, 0\} \\
= &\min_{\{p_0\geq 0,(x_{ik},y_{ij})\in X\}} \Gamma p_0 + \sum_{i=1}^{N} \sum_{j=i+1}^{N} w_{ij}y_{ij} + \sum_{(i,j)\in J} y_{ij}\max\{\hat{w}_{ij} - p_0, 0\},
\end{aligned} \tag{11}$$

where the equality is obtained by the fact $y_{ij}$ is binary in the feasible set $X$.

By the composition $[0,e_1],[e_1,e_2],\cdots,[e_{|J|},\infty)$ of $[0,\infty)$ for $p_0$, we have

$$\sum_{(i,j)\in J} y_{ij}\max\{\hat{w}_{ij} - p_0, 0\} = \begin{cases} \sum_{(i,j):\hat{w}_{ij}\geq \hat{w}_{i^l,j^l}} (\hat{w}_{ij} - p_0)y_{ij}, & \text{if } p_0 \in [e_{l-1}, e_l], l = 1,\cdots,|J|; \\ 0, & \text{if } p_0 \in [e_{|J|},\infty). \end{cases}$$

Thus, the optimal objective value of (6) is $\min_{l=1,\cdots,|J|,|J|+1}\{Z^l\}$, where

$$Z^l = \min_{\{p_0\in[e_{l-1},e_l],(x_{ik},y_{ij})\in X\}} \left( \Gamma p_0 + \sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{ij}y_{ij} + \sum_{(i,j):\hat{w}_{ij}\geq \hat{w}_{i^l,j^l}} (\hat{w}_{ij}-p_0)y_{ij} \right), \quad (12)$$

for $l = 1,\cdots,|J|$, and

$$Z^{|J|+1} = \min_{\{p_0\geq e_{|J|},(x_{ik},y_{ij})\in X\}} \Gamma p_0 + \sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{ij}y_{ij}.$$

For $l = 1,\cdots,|J|$, since the objective function (12) is linear over the interval $p_0 \in [e_{l-1},e_l]$, the optimal is either at the point $p_0 = e_{l-1}$ or $p_0 = e_l$. For $l = |J|+1$, $Z^l$ is obtained at the point $e_{|J|}$ since $\Gamma \geq 0$.

Thus, the optimal value $\min_{l=1,\cdots,|J|,|J|+1}\{Z^l\}$ with respect to $p_0$ is obtained among the points $p_0 = e_l$ for $l = 0,1,\cdots,|J|$. Let $G^l$ be the value at point $p_0 = e_l$ in (12), i.e.,

$$G^l = \Gamma e_l + \min_{(x_{ik},y_{ij})\in X} \left\{ \sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{ij}y_{ij} + \sum_{(i,j):\hat{w}_{ij}\geq e_{l+1}} (\hat{w}_{ij}-e_l)y_{ij} \right\}.$$

We finish the proof. □

As shown in Theorem 2, $G^{|J|} = \Gamma e_{|J|} + \sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{ij}y_{ij}$ is the original nominal problem. Our Algorithm is based on this theorem.

| **Algorithm** |
|---|
| Step 1: For $l = 0,1,\cdots,|J|$, solving $G^l$ in (10); |
| Step 2: Let $l^* = \arg\min_{l=0,1,\cdots,|J|} G^l$; |
| Step 3: Then $\{x^*_{ik},y^*_{ij}\} = \{x_{ik},y_{ij}\}^{l^*}$. |

Similarly, for the robust critical node problem, we have the following theorem, and we omit the proof here.

**Theorem 3.** *Solving robust critical node problem* (9) *is equivalent to solve the* $|J|+1$ *problems* $H^l$*s for* $l = 0,1,\cdots,|J|$, *where the problem* $H^l$ *is formulated as follows:*

$$H^l = \Gamma e_l + \min_{(v_i,u_{ij})\in Y} \left\{ \sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{ij}u_{ij} + \sum_{(i,j):\hat{w}_{ij}\geq e_{l+1}} (\hat{w}_{ij}-e_l)u_{ij} \right\}. \quad (13)$$

In the case of $\Gamma = 0$, which means none of $w_{ij}$s is allowed to change, both RGPP and RCNP become nominal problems (2) and (4), respectively.

## 3 Networks Analysis by GPP and CNP

Before presenting the relations of GPP, CNP and networks, we first introduce the two important properties in networks: community structure and influential individuals. As mentioned in [9], many systems take the form of networks, such as co-author networks, telecommunication networks, supply chain networks, Internet and Worldwide Web, power grids, networks in social society, as well as many biological networks, including neural networks, food webs and metabolic networks. Two well-known networks that have been studied much are scale-free networks, which mean the degree distribution of them follows a power law, and small-world networks, or known as six degrees of separation.

If a network can be divided into several groups such that the nodes within a group have denser connections than nodes from different groups, this network is said to have **community structure**. A **community** is the group of nodes in such division. For example, in a co-author network, the groups may mean different research scientists under different research topics. Mathematically, in a graph $G = (V, E)$ for a network, the vertex set can be divided according to the weights between them into several subsets such that the vertices within a subset have heavier sum weights of edges than that among distinct subsets. The partitioning of graph is exactly the process to detect community structure in a network while the subsets are corresponding to communities. As we have mentioned above, the graph partitioning is NP-complete and is hard to solve. Thus, community detection in a network, especially for arising complex networks for some real world problems, is a difficult task.

Despite of these difficulties, several methods for community detection have been developed and employed with application in different areas successfully, such as minimum-cut method [11], hierarchical clustering, Girvan-Newman algorithm [9], modularity maximization [12], and clique percolation method [13]. Our proposed method graph partitioning can be considered as a generalization of minimum-cut method with more than two subsets. It is different from clique percolation method where each subset is a complete graph. Although these methods work in different situations, we still have the question such that how many possible groups we have, and what the sizes of groups are in a network. These are the parameters $K$ and $n_1, \cdots, n_K$ required in graph partitioning. In graph partitioning, the paper [6] has discussed how to obtain $K$ and presented that the cardinalities $n_1, \cdots, n_K$ can be relaxed to an interval region $[C_{min}, C_{max}]$.

In fact, the constraint of the sum $\sum_{k=1}^{K} x_{ik} = 1$ requires each vertex is partitioned into one and only one subset. The $\sum_{i=1}^{N} x_{ik} = n_k$ defines the size of the $k$th subset. Since $x_{ik} \in \{0, 1\}$, the sum $\sum_{i=1}^{N} x_{ik}$ takes integer values between the lower size bound $C_{min}$ and the upper bound $C_{max}$. These two bounds are known parameters and can be chosen roughly from $\{1, \cdots, N\}$. The two kinds of constraints ensure that each vertex belongs to exactly one subset and all vertices have corresponding subsets. The later one is guaranteed by the fact that $\sum_{k=1}^{K} n_k = \sum_{k=1}^{K} \sum_{i=1}^{N} x_{ik} = \sum_{i=1}^{N} \sum_{k=1}^{K} x_{ik} = \sum_{i=1}^{N} 1 = N$, which means that $n_k$ can take any integer values in $[C_{min}, C_{max}]$ but their sum is fixed as $N$. Thus, in analyzing networks, we only need the information about the rough region of the sizes for communities.

The weight of an edge in a graph describes the closeness of the connections between two nodes in a network. However, the weight is not always certain. It has dynamic changes along the time. The proposed robust optimization model of graph partitioning is to deal with such uncertainty.

Another property arising in networks is the **influential individuals**, which represent the most important nodes in networks. For example, in a co-author network, some scientists may have bigger contributions to or influences in the research society, and they can be considered as influential nodes in this network. In a supply chain network, some logistics centers have the most important positions for satisfying the supply and demand in the supply chain and these centers are the influential nodes. Obviously, detecting these nodes are important for logistics companies so that they can design emergency plans early before having possible destroy or other problems arising in these centers. Different from wide studies in community structures in networks, the research in influence individuals is rare.

In graph theory, the centrality is used to determine the relative importance of a vertex within the graph [7,14]. Four measures of centrality include degree centrality, betweenness, closeness and eigenvector centrality. Different from the methods used in [10,17], we use the critical node detection methods in graphs [1] and the importance of vertices is measured based on the connectivity. A node is said to have influence in the network if deletion of it results in the maximum number of disconnect components in the network. Similarly, the weights between vertices are always uncertain, and we also construct the robust optimization model of critical node detection.

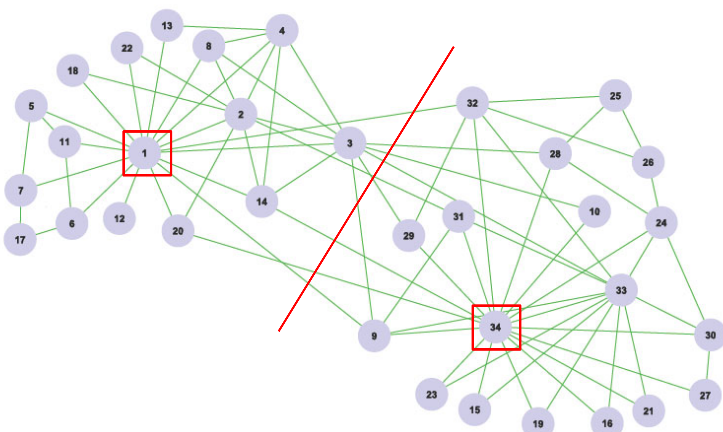


**Fig. 1.** Zachary's karate club with two communities and two influential nodes: 1, 34

When concentrating on some networks, we always find that several dense groups appear and these groups always have centers within them. Using the concepts mentioned above, we say that this is the phenomenon of community with influence individual center. This is the reason why we use both GPP and CNP to analyze the networks in the meantime.

Given a network with the graph notation $G = (V,E)$ with the weight matrix $W$ to measure the closeness of connectivity between nodes, we are deciding to detect $K$ communities and also $K$ influential individuals in the network.

In addition, we can study the influential individuals within each community. After obtaining the community structure of the network, we use the the CNP model on each community and the two or three influential individuals with each community can be found. These parameters $K(K \in \{2,\cdots,N-1\}), n_1,\cdots,n_K$ for analyzing are obtained from experiences or direct observations. For robust models, we assume that every weight $w_{ij}$ of edge $(v_i, v_j)$ has the uncertainty which is a more close description of real networks. In next section, we discuss several numerical experiments on networks.

## 4 Numerical Experiments

In this section, we consider a well-known social networks: Zachary's karate club [15] and a generated artificial network with uncertainties on edges. For the network of Zachary's karate club, we use GPP and CNP models to identify the two groups in this club and the leaders, respectively. For the artificial network, we use robust models of GPP and CNP to study its community structures and influential individuals. The algorithms based on our models are implemented using CPLEX 11.0 via ILOG Concert Technology 2.5, and all computations are performed on a SUN UltraSpace-III with a 900 MHz processor and 2.0 GB RAM. Computational times are reported in CPU seconds.

The Zachary's karate club [15] is a network consists of 34 nodes and 78 links representing friendships between members of the club over two years' period. In the study of this network [15], a disagreement, between administrator of the club and the club's instructor, resulted two groups. One group is the members leaving with the instructor to start a new club. The corresponding parameters are $K = 2, n_1 = 16, n_2 = 18$ in our models of GPP and CNP. It shows that our algorithm based on GPP model can find the two groups correctly as shown in Fig. 1, which is the same as that in the study of [9]. In addition, by our model of CNP, we correctly find the two influential individuals with in two groups: instructor (node 1) and the administrator (node 34). If assume $K = 5$ in our CNP model, we can find influential nodes 1, 3, 2 in the first group and nodes 34, 33 in the second group, which are named as core vertices in [9].

In the following, we use Matlab to generate an artificial network with 22 nodes and 48 links with weight in interval ranges (see Table 1). The values of $w_{ij}$ and $\hat{w}_{ij}$ in

**Table 1.** Uncertainties of 48 links

| Edge | $[l_{ij}, u_{ij}]$ | Edge | $[l_{ij}, u_{ij}]$ | Edge | $[l_{ij}, u_{ij}]$ | Edge | $[l_{ij}, u_{ij}]$ |
|---|---|---|---|---|---|---|---|
| ( 1, 2 ) | [ 0.11, 0.67 ] | ( 4, 10 ) | [ 0.62, 1.54 ] | ( 6, 17 ) | [ 1.49, 2.97 ] | ( 9, 20 ) | [ 0.21, 3.01 ] |
| ( 1, 3 ) | [ 0.07, 0.49 ] | ( 4, 14 ) | [ 0.78, 0.98 ] | ( 6, 19 ) | [ 0.79, 0.95 ] | ( 10, 20 ) | [ 0.69, 0.71 ] |
| ( 1, 6 ) | [ 0.80, 3.50 ] | ( 4, 20 ) | [ 0.34, 1.78 ] | ( 6, 21 ) | [ 0.43, 0.93 ] | ( 11, 20 ) | [ 0.36, 1.48 ] |
| ( 1, 7 ) | [ 0.50, 1.28 ] | ( 5, 13 ) | [ 0.29, 1.15 ] | ( 7, 10 ) | [ 0.35, 0.79 ] | ( 12, 13 ) | [ 0.35, 2.01 ] |
| ( 1, 17 ) | [ 0.65, 2.17 ] | ( 5, 14 ) | [ 1.35, 0.53 ] | ( 7, 18 ) | [ 0.01, 1.03 ] | ( 12, 19 ) | [ 0.78, 3.70 ] |
| ( 1, 18 ) | [ 0.29, 0.22 ] | ( 5, 15 ) | [ 0.45, 1.73 ] | ( 7, 20 ) | [ 0.30, 0.66 ] | ( 12, 22 ) | [ 1.04, 2.12 ] |
| ( 1, 19 ) | [ 0.08, 0.22 ] | ( 5, 16 ) | [ 0.86, 3.32 ] | ( 8, 11 ) | [ 1.37, 0.63 ] | ( 13, 14 ) | [ 0.40, 0.90 ] |
| ( 2, 11 ) | [ 0.71, 1.41 ] | ( 5, 19 ) | [ 0.94, 1.18 ] | ( 8, 13 ) | [ 0.22, 0.44 ] | ( 13, 21 ) | [ 0.39, 0.59 ] |
| ( 2, 20 ) | [ 1.21, 1.63 ] | ( 5, 21 ) | [ 0.94, 2.20 ] | ( 8, 20 ) | [ 0.14, 2.20 ] | ( 14, 15 ) | [ 0.79, 2.63 ] |
| ( 3, 8 ) | [ 0.27, 1.69 ] | ( 5, 22 ) | [ 0.70, 1.16 ] | ( 9, 11 ) | [ 0.67, 3.41 ] | ( 15, 16 ) | [ 0.50, 0.66 ] |
| ( 3, 9 ) | [ 0.42, 1.40 ] | ( 6, 7 ) | [ 0.51, 0.77 ] | ( 9, 13 ) | [ 0.07, 0.17 ] | ( 17, 18 ) | [ 0.56, 1.26 ] |
| ( 4, 5 ) | [ 0.76, 0.86 ] | ( 6, 12 ) | [ 0.48, 0.80 ] | ( 9, 15 ) | [ 0.51, 0.71 ] | ( 21, 22 ) | [ 0.84, 4.24 ] |

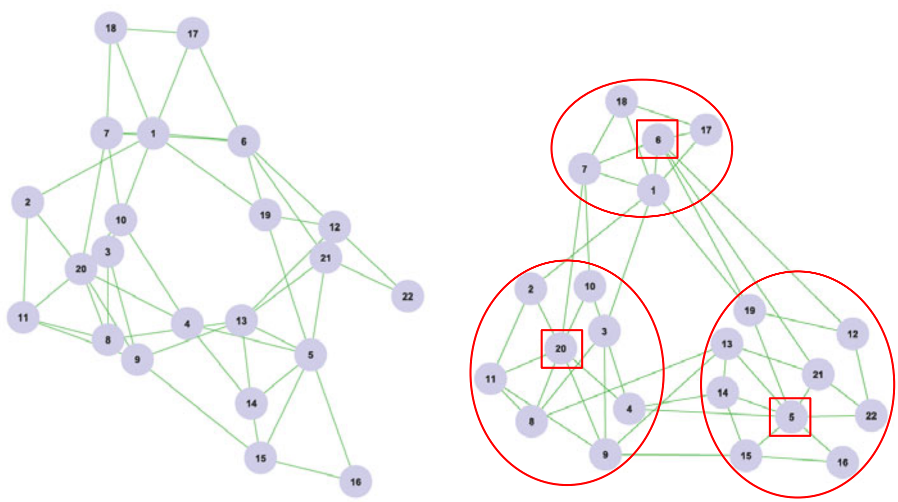


**Fig. 2.** An artificial network with 22 nodes and 48 links

problems (5) and (8) can be easily computed from the interval values $[l_{ij}, u_{ij}]$. We use robust optimization models of GPP and CNP to analyze this network (see Fig. 2). In this network, 3 communities are found with 5 nodes, 8 nodes and 9 nodes respectively. Assume $K = 3$ and $\Gamma = 48$ in RCNP, three influential nodes (nodes 5, 6, and 20) are found, and each of them is in a community. Observing the node 1 and node 6 in the same group, there are 6 edges incident with node 6 and 7 edges incident with node 1. The weighted degree for node 6 is in $[4.50, 9.92]$, while the weighted degree for node 1 is in $[2.50, 8.55]$. Node 6 is chosen as the influential node in this group in the worst cases ($4.50 > 2.50$ and $9.92 > 8.55$).

Next, we compare the methods proposed in Section 2.3 with the direct method in CPLEX to solve the equivalent formulations (6) and (9). In Table 2, we present the computational seconds and computational results. The methods in Theorem 2 and

**Table 2.** Comparisons of two computational methods

| Graphs and Parameters | | | | | | | RGPP | | | | RCNP | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | CPLEX (6) | | Method Thm 2 | | CPLEX (9) | | Method Thm 3 | |
| $N$ | $r$ | $[l,u]$ | $\|J\|$ | $\Gamma$ | $K$ | $n_1, \cdots, n_K$ | Seconds | Results | Seconds | Results | Seconds | Results | Seconds | Results |
| 10 | 0.1 | [0,1] | 4 | 2 | 3 | 3,3,4 | 0 | 0 | 0 | 0 | 0.04 | 0 | 0.08 | 0 |
| | 0.2 | | 9 | 5 | 3 | 3,3,4 | 0.08 | **2.55** | 0.03 | **2.55** | 0.03 | **2.46** | 0.02 | **2.46** |
| | 0.3 | | 13 | 7 | 3 | 3,3,4 | 0.13 | **6.29** | 0.10 | 6.55 | 0.03 | **4.22** | 0.12 | **4.22** |
| 20 | 0.1 | [0,1] | 19 | 10 | 4 | 4,5,5,6 | 0.36 | **6.39** | 0.28 | 7.07 | 0.65 | **8.66** | 0.41 | 9.46 |
| | 0.2 | | 37 | 19 | 4 | 4,5,5,6 | 9.07 | **22.10** | 5.05 | **22.10** | 1.58 | **26.30** | 1.66 | 28.63 |
| | 0.3 | | 57 | 29 | 4 | 4,5,5,6 | 35.93 | **43.25** | 14.37 | 43.77 | 1.31 | **41.16** | 1.35 | 41.16 |
| 30 | 0.1 | [0,1] | 42 | 22 | 4 | 5,7,8,10 | 10.35 | **16.38** | 2.02 | 16.58 | 18.23 | **31.52** | 6.87 | 33.27 |
| 40 | 0.1 | [0,1] | 78 | 39 | 4 | 8,9,10,13 | 153.67 | **33.85** | 194.44 | **33.85** | 187.42 | 78.27 | 49.22 | **78.02** |
| 50 | 0.1 | [0,1] | 122 | 61 | 4 | 9,12,13,16 | >3000 | **65.46** | 2167.83 | 67.43 | 668.66 | **118.98** | 398.00 | 120.26 |

**Table 3.** Subsets and critical nodes

| Graphs | RGPP | RCNP |
|---|---|---|
| 10 0.1 | (2,3,5) | |
| | (4,7,9) | 9 |
| | (1,6,8,10) | 6,10 |
| 10 0.2 | (1,2,9) | 9 |
| | (4,7,10) | 5 |
| | (3,5,6,8) | 10 |
| 10 0.3 | (2,3,6) | |
| | (1,5,10) | 5 |
| | (4,7,8,9) | 7,9 |
| 20 0.1 | (3,4,6,12) | |
| | (2,8,9,13,17) | 8 |
| | (5,11,14,18,19) | 14,18 |
| | (1,7,10,15,16,20) | 1 |
| 20 0.2 | (10,16,17,19) | |
| | (4,8,11,14,18) | 4,18 |
| | (1,3,7,13,15) | 1 |
| | (2,5,6,9,12,20) | 12 |
| 20 0.3 | (5,6,13,19) | 13 |
| | (2,3,7,10,16) | 2 |
| | (8,9,12,14,18) | |
| | (1,4,11,15,17,20) | 4,11 |
| 30 0.1 | (11,17,20,21,29) | |
| | (4,10,13,14,16,19,23) | 13 |
| | (1,6,7,9,12,22,27,28) | 12 |
| | (2,3,5,8,15,18,23,24,26,30) | 8,18 |
| 40 0.1 | (4,7,11,15,18,28,29,40) | |
| | (1,3,6,8,12,2,4,30,33,39) | 39 |
| | (2,5,9,10,14,16,20,21,25,38) | 5 |
| | (13,17,19,22,23,26,27,31,32,34,35,36,37) | 19,23 |
| 50 0.1 | (4,8,18,20,21,28,30,33,40) | 21 |
| | (1,9,12,17,22,24,25,26,37,41,42,43) | |
| | (2,6,11,13,15,16,19,23,31,45,46,48,49) | |
| | (3,5,7,10,14,27,29,32,34,35,36,38,39,44,47,50) | 35,38,50 |

Theorem 3 are also implemented in ILOG Concert Technology 2.5. The gaps for all these methods in CPLEX are set as 0.1. The parameter $r$ in a graph is the density, which is the ratio of the number of edges and the number of possible edges.

From Table 2, for robust graph partitioning problems, the method by Theorem 2 is more efficient than default CPLEX method (6) in most cases; for robust critical node problem, the method by Theorem 3 is also more efficient than default CPLEX method (9) in most cases. The better results are in bold. In Table 3, we present the subsets and critical nodes from the better results. For $K$ critical nodes and $K$ subsets, in many cases, the $K$ critical nodes are distributed in $K-1$ subsets.

## 5 Conclusions

In this paper, for a given network modeled by a graph model with certain and uncertain weights of links, we have presented the optimization models for both graph partitioning problem and critical node problem. These models are formulated as binary integer linear programs. An algorithm based on a decomposition method on one variable is presented. After that, we introduce two important structures: community and influential individuals in networks. We have established the relationship between these two structures with GPP and CNP in graph theory.

Because of the uncertainties arising in real social or biological networks, the robust optimization models of GPP and CNP are quite useful to analyze these complex networks. We also present several numerical experiments to analyze the networks by RGPP and RCNP. It shows our models are quite useful. However, because of the NP-completeness of the nominal problems GPP and CNP, the RGPP and RCNP are quite complex in computation. Designing efficient algorithms for such problems is still under discussion. On the other hand, since the real networks in practice are always random and have some properties, such as scale-free and small-world, the further research can concentrate on combining such problems with these propositions.

Moreover, a set of critical nodes has some specific functions in some networks with dynamic situations. For example, the network modeling for epileptic brain is constructed by nonlinear dynamic measurements [16]. Part of the brain has special functions to control the movement of body. This network can be analyzed by our proposed method to find the critical sites of such functional nodes.

## References

1. Arulselvan, A., Commander, C.W., Elefteriadou, L., Pardalos, P.M.: Detecting critical nodes in sparse graphs. Computers and Operations Research 36(7), 2193–2200 (2009)
2. Arulselvan, A., Commander, C.W., Pardalos, P.M., Shylo, O.: Managing network risk via critical node identification. In: Gulpinar, N., Rustem, B. (eds.) Risk Management in Telecommunication Networks. Springer, Heidelberg (2010)
3. Bertsimas, D., Sim, M.: The price of robustness. Operations Research 52(1), 35–53 (2004)
4. Boginski, V., Commander, C.W.: Identifying critical nodes in protein-protein interaction networks. In: Butenko, S.I., Chaovilitwongse, W.A., Pardalos, P.M. (eds.) Clustering Challenges in Biological Networks, pp. 153–167. World Scientific, Singapore (2008)
5. Fan, N., Pardalos, P.M., Chinchuluun, A., Pistikopoulos, E.N.: Graph partitioning approaches for analyzing biological networks. In: Mondaini, R.P. (ed.) BIOMAT 2009 International Symposium on Mathematical and Computational Biology, pp. 250–262. World Scientific, Singapore (2010)
6. Fan, N., Pardalos, P.M.: Linear and quadratic programming approaches for the general graph partitioning problem. Journal of Global Optimization 48(1), 57–71 (2010)
7. Freeman, L.C.: A set of measures of centrality based on betweenness. Sociometry 40, 35–41 (1977)
8. Garey, M.R., Johnson, D.S., Stockmeyer, L.: Some simplified NP-complete graph problems. Theor. Comput. Sci. 1, 237–267 (1976)
9. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proc. Natl. Acad. Sci. 99, 7821–7826 (2002)

10. Kempe, D., Kleinberg, J., Tardos, E.: Influential nodes in a diffusion model for social networks. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 1127–1138. Springer, Heidelberg (2005)
11. Newman, M.E.J.: Detecting community structure in networks. Eur. Phys. J. B 38, 321–330 (2004)
12. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. Phys. Rev. E 69, 066133 (2004)
13. Palla, G., Derenyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature 435, 814–818 (2005)
14. Sabidussi, G.: The centrality index of a graph. Psychometrika 31(4), 581–603 (1966)
15. Zachary, W.W.: An information flow model for conflict and fission in small groups. Journal of Anthropological Research 33, 452–473 (1977)
16. Zhang, J., et al.: Real-time differentiation of nonconvulsive status epilepticus from other encephalopathies using quantitative EEG analysis: a pilot study. Epilepsia 51(2), 243–250 (2010)
17. Zou, F., Zhang, Z., Wu, W.: Latency-bounded minimum influential node selection in social networks. In: Liu, B., Bestavros, A., Du, D.-Z., Wang, J. (eds.) Wireless Algorithms, Systems, and Applications. LNCS, vol. 5682, pp. 519–526. Springer, Heidelberg (2009)