Identifying risk-averse low-diameter clusters in graphs with stochastic vertex weights

Maciej Rysz[1] Foad Mahdavi Pajouh[2] Pavlo Krokhmal[3]* Eduardo L. Pasiliao[4]

[1]National Research Council
Air Force Research Labratory
101 West Eglin Blvd, Eglin AFB, FL 32542
E-mail: mwrysz@yahoo.com

[2]Management Science and Information Systems Department
University of Massachusetts Boston
100 Morrissey Blvd., Boston, MA 02125
E-mail: foad.mahdavi@umb.edu

[3]Department of Systems and Industrial Engineering
University of Arizona, 1127 E James E. Rogers Way, Tucson, AZ 85721
E-mail: krokhmal@email.arizona.edu

[4]Munitions Directorate
Air Force Research Laboratory
101 West Eglin Blvd, Eglin AFB, FL 32542
E-mail: eduardo.pasiliao@eglin.af.mil

Abstract

In this work, we study the problem of detecting risk-averse low-diameter clusters in graphs. It is assumed that the clusters represent k-clubs and that uncertain information manifests itself in the form of stochastic vertex weights whose joint distribution is known. The goal is to find a k-club of minimum risk contained in the graph. A stochastic programming framework that is based on the formalism of coherent risk measures is used to quantify the risk of a cluster. We show that the selected representation of risk guarantees that the optimal subgraphs are maximal clusters. A combinatorial branch-and-bound algorithm is proposed and its computational performance is compared with an equivalent mathematical programming approach for instances with $k = 2, 3$, and 4.

Keywords: k-club; low-diameter clusters; stochastic graphs; coherent risk measures; combinatorial branch-and-bound

*Corresponding author

1 Introduction

Graphs are effective tools for modeling many real-world systems and the complex interactions between their components. A typical graph model assigns vertices to represent a system's components and a set of edges to describe the connections and/or relationships between them. Well-known examples of such frameworks are represented by many systems studied in social network analysis, transportation, telecommunications, computational finance, and so on. Additionally, graph-based data mining methods [18] provide powerful techniques for analyzing and understanding systems whose descriptive data may be represented using a graph.

A principal application of graph-based data mining involves the identification of subgraphs, referred to as clusters, corresponding to subsystems with a given structural or functional property. For example, in social networks, detecting highly-connected clusters can be used for advertising and marketing purposes [22, 23, 46]; in stock market graphs, it can be used for identifying diverse portfolios [12]; and in call graphs, it can be used for detecting communicating clusters [1].

One of the basic problems in this context entails finding the largest "perfectly" cohesive group within a network such that the confined members are all interconnected, also known as the maximum clique (complete subgraph) problem. Several prominent studies provided the basis for exact combinatorial solution algorithms for the maximum clique problem [8, 16, 33]. In particular, Carraghan and Pardalos [16] introduced a recursive branch-and-bound method for finding maximum cliques by exploiting the heredity property [42] of complete subgraphs. Subsequent extensions of their work enhanced the process of reducing solution space via vertex coloring schemes for estimation of upper bounds on the maximum achievable subgraph sizes during the branch-and-bound procedure (e.g. [15, 24, 41]).

In many practical applications, the requirement that the desired subgraph must be complete may, however, impose excessive restrictions and therefore warrant some structural relaxation in terms of cluster connectivity. As a consequence, a number of clique relaxation models have been proposed in graph theory literature, which relax the completeness property relative to the degree of the member vertices, their distance from each other, or the density of the subgraph. A comprehensive review of clique relaxation models is provided in [10]. In the present work, we focus on a specific type of clique relaxation, known as the *k-club* [3], which represents a subgraph whose members are connected via at most $k-1$ intermediary members. The k-club model effectively represents low-diameter clusters that may reveal valuable information embedded in social, financial, and telecommunication networks. Several recent studies proposed combinatorial branch-and-bound methods and presented complexity results associated with finding maximum k-clubs in graphs [13, 17, 34, 39].

An important extension of the described class of problems involves the imposition of topologically exogenous information in the form of deterministic vertex weights, and correspondingly finding a subset of maximum weight that conforms to a defined structural property. Similar exact weight-based branch-and-bound solution techniques have been developed for determining the maximum-weight subgraphs [7, 28, 32].

Numerous circumstances may further justify the imposition of uncertain exogenous information over the graph's edges that influences network flow distribution, robustness, and costs [4, 6, 14, 20, 21, 44]. However, far fewer endeavors consider decision making relative to the optimal allocation of resources over defined subgraph topologies when uncertainties are induced by stochastic factors associated with network vertices [38]. For example, in social networks or call graphs, the uncertainties related to the value or reliability of the information provided by each entity can be modeled by random weights on vertices whose relationships or communications are presented by edges. Similarly, in stock market graphs, the uncertainties associated with returns on investments from different assets can be defined as random weights assigned to their corresponding vertices, with edges linking highly correlated assets (vertices).

In this study, we extend the techniques introduced in [38] to address problems of finding subgraphs of minimum risk that represent a k-club. A probabilistic framework utilizing the distributional information of stochastic vertex weights by means of *coherent measures of risk* [5, 19] is employed to define a *risk-averse k-club* (RA-k) problem as finding the lowest risk k-club in a network. As an illustrative example, we focus on instances when $k = 2, 3, 4$, and utilize a mathematical programming formulation introduced in [43] for finding a maximum k-club in a graph. A combinatorial branch-and-bound method for finding a largest k-club [13, 17, 34] is also modified to accommodate the conditions of RA-k problem via risk-based branching and bounding schema. We compare the solution performance of the proposed branch-and-bound algorithm relative to solving the mathematical programming formulation for the RA-k problem using a state-of-the-art commercial solver.

The remainder of the paper is organized as follows. In Section 2, we examine the general representation of RA-k problem and discuss its properties. Section 3 presents a mathematical programming formulation and a combinatorial branch-and-bound method for solving the RA-k problem. Finally, Section 4 furnishes numerical studies demonstrating the computational performances of the developed branch-and-bound method and the aforementioned mathematical programming approach on problems where risk is quantified using higher-moment coherent risk measures [27].

2 Risk-averse k-club problem

Given an undirected graph $G = (V, E)$ and any subset of its vertices $S \subseteq V$, let $G[S]$ represent the subgraph of G induced by S, i.e., $G[S] = (S, E \cap (S \times S))$. Let $\mathcal{Q}$ denote the desired property which the induced graph $G[S]$ must satisfy. The present work considers the case when $\mathcal{Q}$ represents a certain relaxation of the *completeness* property, such that a subgraph with property $\mathcal{Q}$ represents a *clique relaxation*.

Depending on the characteristic of a complete graph that is relaxed, clique relaxations can be categorized into *density-based* [1, 2, 35], *degree-based* [40], and *distance-based* [3, 29, 30] relaxations. In this work, property $\mathcal{Q}$ represents a special distance-based relaxation of the completeness property. For a formal definition, let $d_G(i, j)$ denote the distance between nodes $i, j \in V$ in graph G, measured as the number of edges in a shortest path between i and j in G. Then, a subset of vertices $S \subseteq V$ of graph G is called a *k-clique* if

$$\max_{i,j \in S} d_G(i, j) \leq k.$$

Note that the definition of the k-clique does not require that the shortest path between $i, j \in S$ belong to $G[S]$. If one requires that the shortest path between any two vertices i, j in S belong to the induced subgraph $G[S]$, then the subset S such that

$$\max_{i,j \in S} d_{G[S]}(i, j) \leq k, \tag{1}$$

is called a *k-club*. Note that a k-club is also a k-clique, while the inverse is not true in general. By definition, 1-cliques and 1-clubs are cliques. Throughout the remainder of this study, we let $\Gamma_G(k)$ denote the set of all k-clubs in graph G:

$$\Gamma_G(k) = \big\{S \subseteq V : d_{G[S]}(i, j) \leq k,\ \forall i, j \in S\big\}. \tag{2}$$

Additionally, a k-club is said to be maximal, if it is not strictly contained in another k-club; and a maximum k-club is a k-club of the largest order in graph G.

A popular class of graph-theoretical problems is represented by the *maximum weight subgraph* problems, which are concerned with finding a subset S of vertices in G such that the induced subgraph satisfies the given property $\mathcal{Q}$ and has the largest weight (defined as the sum of its vertices' weights). The *maximum weight k-club* problem is then formulated as

$$\max\Big\{\sum_{i\in S} w_i \,:\, S \in \Gamma_G(k)\Big\}, \tag{3}$$

where $w_i \geq 0$ represents the weight of vertex i and the set $\Gamma_G(k)$ is defined by (2). Clearly, an optimal set S in problem (3) will be *maximal*, but not necessarily *maximum* (of the largest order) set with property $\mathcal{Q}$. If the weight of each vertex is one, the maximum weighted k-club problem is simply referred to as the maximum k-club problem.

In this work, we consider an extension of problem (3) that assumes stochastic vertex weights. In this case, a direct translation into a stochastic framework is not straightforward due to the fact that the maximization of random weights would be ill-posed in context of stochastic programming resulting from the absence of a deterministic optimal solution. Likewise, maximization of the expected weight of the sought set is rather uninteresting in the sense that it reduces to the deterministic version of the problem presented above. A more suitable approach, thus, involves computing the subgraph's weight via a (nonlinear) statistical function that utilizes the distributional information about the weights' uncertainties, rather than a simple sum of its vertices' stochastic weights. In particular, we pursue a *risk-averse* approach so as to find the subgraph of G that has the *lowest risk* and satisfies property $\mathcal{Q}$. Let X_i denote a random variable that represents the costs of losses associated with vertex $i \in V$, such that the joint distribution of vector $\mathbf{X}_G = (X_1, \ldots, X_{|V|})$ is known. Then, the problem of finding the *minimum risk* subgraph in G that has property $\mathcal{Q}$, or the *risk-averse $\mathcal{Q}$ problem* takes the form:

$$\min\big\{\mathscr{R}(S;\mathbf{X}_G) : S \subseteq V \text{ and } G[S] \text{ satisfies } \mathcal{Q}\big\}, \tag{4}$$

where $\mathscr{R}(S;\mathbf{X}_G)$ is the risk associated with set S given the distributional information $\mathbf{X}_G$. In the particular case when property $\mathcal{Q}$ ensures that the subgraph in question is a k-club, formulation (4) defines the *risk-averse k-club problem* (RA-k),

$$\min\big\{\mathscr{R}(S;\mathbf{X}_G) : S \in \Gamma_G(k)\big\}, \tag{5}$$

which represents a risk-averse stochastic generalization of the deterministic maximum weight k-club problem (1), as shown below.

A constructive form of risk function $\mathscr{R}(S;\mathbf{X}_G)$ can be introduced by employing the well-known in stochastic optimization literature concept of *risk measure* [26]. Given a probability space $(\Omega, \mathcal{F}, \mathsf{P})$, where Ω is the set of random events, $\mathcal{F}$ is the σ-algebra, and P is a probability measure, a risk measure ρ is defined as a mapping $\rho : \mathcal{X} \mapsto \mathbb{R}$, where $\mathcal{X}$ is a linear space of $\mathcal{F}$-measurable functions $X : \Omega \mapsto \mathbb{R}$. In what follows, the space $\mathcal{X}$ is assumed to possess the properties necessary for the risk measures introduced below to be well-defined. Namely, $\mathcal{X}$ is supposed to allow for a sufficient degree of integrability, in particular, $\mathsf{E}|X| < \infty$, and be endowed with an appropriate topology, e.g., the topology induced by convergence in probability. Lastly, we consider risk measures that are *proper* functions on $\mathcal{X}$, i.e., $\rho(X) > -\infty$ for all $X \in \mathcal{X}$ and $\{X \in \mathcal{X} : \rho(X) < \infty\} \neq \emptyset$.

Then, assuming that risk measure ρ is lower semi-continuous (l.s.c.), the risk $\mathscr{R}(S;\mathbf{X}_G)$ of a set $S \subseteq V$ with uncertain vertex weights X_i, $i \in V$, can be defined as the optimal value of the following stochastic programming problem:

$$\mathscr{R}(S;\mathbf{X}_G) = \min\Big\{\rho\Big(\sum_{i\in S} u_i X_i\Big) : \sum_{i\in S} u_i = 1;\ u_i \geq 0,\ i \in S\Big\}. \tag{6}$$

Note that this definition of the set risk function $\mathscr{R}(\cdot)$ admits risk reduction through diversification as illustrated by the following proposition:

Proposition 1 ([38]) *Given a graph $G = (V, E)$ with stochastic weights X_i, $i \in V$, and a l.s.c. risk measure ρ, the set risk function $\mathscr{R}$ defined by* (6) *satisfies*

$$\mathscr{R}(S_2; \mathbf{X}_G) \leq \mathscr{R}(S_1; \mathbf{X}_G) \quad \textit{for all} \quad S_1 \subseteq S_2. \tag{7}$$

The following observation regarding the optimal solution of the risk-averse $\mathcal{Q}$ problem (4) stems directly from property (7):

Corollary 1 *There exists an optimal solution of the risk-averse $\mathcal{Q}$ problem* (4) *with $\mathscr{R}(S; \mathbf{X}_G)$ defined by* (6) *that is a maximal set with property $\mathcal{Q}$ in G.*

Additional properties of $\mathscr{R}(S; \mathbf{X}_G)$ as defined by (6) ensue from the assumption that the risk measure ρ belongs to the family of coherent measures of risk [5], i.e., satisfies the properties of monotonicity, $\rho(X) \leq 0$ for all $X \leq 0$; subadditivity, $\rho(X + Y) \leq \rho(X) + \rho(Y)$; transitional invariance, $\rho(X + c) = \rho(X) + c$ for all $c \in \mathbb{R}$; and positive homogeneity, $\rho(\lambda X) = \lambda\rho(X)$ for all $\lambda > 0$. Then, the corresponding set risk function $\mathscr{R}(S; \mathbf{X}_G)$ satisfies analogous properties with respect to the stochastic weights vector $\mathbf{X}_G$,

(G1) *monotonicity*: $\mathscr{R}(S; \mathbf{X}_G) \leq \mathscr{R}(S; \mathbf{Y}_G)$ for all $\mathbf{X}_G \leq \mathbf{Y}_G$;

(G2) *positive homogeneity*: $\mathscr{R}(S; \lambda\mathbf{X}_G) = \lambda\mathscr{R}(S; \mathbf{X}_G)$ for all $\mathbf{X}_G$ and $\lambda > 0$;

(G3) *transitional invariance*: $\mathscr{R}(S; \mathbf{X}_G + a\mathbf{1}) = \mathscr{R}(S; \mathbf{X}_G) + a$ for all $a \in \mathbb{R}$;

where $\mathbf{1}$ is the vector of ones, and the vector inequality $\mathbf{X}_G \leq \mathbf{Y}_G$ is interpreted component-wise.

Observe that $\mathscr{R}(S; \mathbf{X}_G)$ violates in general the subadditivity requirements with respect to the stochastic weights. However, risk reduction via diversification is guaranteed by (7), which ensures that the inclusion of additional vertices to the existing feasible solution is always beneficial. Further, under the assumption of nonnegative stochastic vertex weights, $\mathbf{X}_G \geq \mathbf{0}$, the set risk $\mathscr{R}(S; \mathbf{X}_G)$ can be shown to be subadditive with respect to subsets of V,

$$\mathscr{R}(S_1 \cup S_2; \mathbf{X}_G) \leq \mathscr{R}(S_1; \mathbf{X}_G) + \mathscr{R}(S_2; \mathbf{X}_G), \quad S_1, S_2 \subseteq V. \tag{8}$$

Clearly, it is required that S_1, S_2, and $S_1 \cup S_2$ satisfy property $\mathcal{Q}$ in conformance to the context of risk-averse $\mathcal{Q}$ problems.

3 Solution approaches for risk-averse k-club problems

In this section, we first address the computational complexity of the RA-k problem for any fixed positive integer k, and show that this problem is $\mathcal{NP}$-hard. We then propose two exact solution algorithms for this problem. First, we consider a mathematical programming approach for the RA-k problem, where the risk $\mathscr{R}(S; \mathbf{X}_G)$ of a set $S \in \Gamma_G(k)$ is defined by (6). To this end, we take advantage of a recent formulation for the maximum k-club problem developed by Veremyev et al. [43]. Next, we propose a combinatorial branch-and-bound algorithm for solving RA-k problem that utilizes the same solution space processing principles for finding maximum k-clubs as the ones used in [13, 17, 34].

In order to establish the problem's complexity and derive the corresponding solution methods, we need to introduce additional assumptions on the properties of stochastic weights $\mathbf{X}_G$ and risk measure ρ involved in the definition of the risk-averse k-club problem (5). Namely, throughout this section it is assumed that the stochastic weights X_i of vertices $i \in V$ are nonnegative and rational-valued, $X_i : \Omega \mapsto \mathbb{Q}_+$, $i \in V$, where $\mathbb{Q}_+$ denotes the set of nonnegative rational numbers. Also, the corresponding probability measure P is rational-valued, i.e., $\mathsf{P}\{X_i = X_i(\omega)\} \in \mathbb{Q}_+ \cap [0,1]$ for all $\omega \in \Omega$ and all $i \in V$. Similarly, we assume that the risk measure ρ is such that $\rho(X) \in \mathbb{Q}$ whenever X and the underlying probability measure are rational-valued. In addition, we restrict our attention to risk measures that are *expectation-bounded*[1] [37], i.e., such that $\rho(X) > \mathsf{E}X$ for all non-constant X, and $\rho(X) = \mathsf{E}X$ for all constant X, or such X that $X = \text{const}$ with probability 1.

3.1 Computational complexity

In this section, we derive the computational complexity of the the risk-averse k-club problem from the complexity of the more general class of risk-averse $\mathcal{Q}$ problems (4).

For a given property $\mathcal{Q}$, the decision version of risk-averse $\mathcal{Q}$ problem, denoted by $\langle G, \mathbf{X}_G, \rho, c\rangle$, is as follows. Given a graph $G = (V, E)$, a vector of stochastic weights $\mathbf{X}_G$, a l.s.c. risk measure ρ, and a $c \in \mathbb{Q}$, determine whether there exists a set $S \subseteq V$ such that $G[S]$ satisfies $\mathcal{Q}$ and $\mathscr{R}(S; \mathbf{X}_G) < c$. We also consider the deterministic maximum $\mathcal{Q}$ problem:

$$\max\{|S| : S \subseteq V \text{ and } G[S] \text{ satisfies } \mathcal{Q}\}, \tag{9}$$

and its decision version, denoted as $\langle G, q\rangle$: given a graph $G = (V, E)$ and an integer q, is there a subset of V that has property $\mathcal{Q}$ and order larger than q?

Theorem 1 *If property $\mathcal{Q}$ is such that the decision version of (deterministic) maximum $\mathcal{Q}$ problem is $\mathcal{NP}$-hard, then the decision version of risk-averse Q problem is also $\mathcal{NP}$-hard, provided that the risk measure ρ is proper, l.s.c., and expectation-bounded.*

Proof: The intractability of the risk-averse $\mathcal{Q}$ problem is proved by a polynomial-time reduction from the maximum $\mathcal{Q}$ problem. Given a graph $G = (V, E)$ and a fixed positive integer q, consider the decision version of the maximum $\mathcal{Q}$ problem $\langle G, q\rangle$. For any such maximum $\mathcal{Q}$ decision problem $\langle G, q\rangle$, we replicate $\hat{G} = G$ and let $\hat{X}_i$ for all $i \in V$ be a set of independently and identically distributed random variables with Bernoulli distribution, such that $\mathsf{P}\{\hat{X}_i = 0\} = \mathsf{P}\{\hat{X}_i = 1\} = \frac{1}{2}$ for all $i \in V$. As a risk measure, we select $\hat{\rho}(X) = \sigma^2(X) + \mathsf{E}X$, where $\sigma^2(X)$ denotes the variance of X. Obviously, $\hat{\rho}(X)$ is expectation bounded, as well as l.s.c. and proper, so that the corresponding set risk function $\mathscr{R}$ is well defined. It is easy to see that the set risk function $\mathscr{R}(S, \hat{\mathbf{X}}_{\hat{G}})$ becomes equal to

$$\begin{aligned}\mathscr{R}(S, \hat{X}_{\hat{G}}) &= \min\Big\{\sigma^2\Big(\sum_{i\in S} u_i \hat{X}_i\Big) + \frac{1}{2} : \sum_{i\in S} u_i = 1;\ u_i \geq 0,\ \forall i \in S\Big\} \\ &= \frac{1}{4|S|} + \frac{1}{2}.\end{aligned}$$

This procedure constructs in polynomial time an instance $\langle G, \hat{\mathbf{X}}_G, \hat{\rho}, \frac{1}{4q} + \frac{1}{2}\rangle$ of risk-averse $\mathcal{Q}$ problem such that there exists a $\mathcal{Q}$-subgraph of order larger than q in G if and only if there exists a $\mathcal{Q}$-subgraph

[1] "Expectation-boundedness" is also known as "aversity" [36], but we use the former term in this work so as to avoid semantic confusion when referring to "risk-averse" subgraphs.

S in G such that $\mathscr{R}(S;\hat{\mathbf{X}}_G) < \frac{1}{4q} + \frac{1}{2}$. This shows that the decision version of risk-averse $\mathcal{Q}$ problem is $\mathcal{NP}$-hard if the maximum $\mathcal{Q}$ problem is $\mathcal{NP}$-hard. $\square$

The computational complexity of RA-k problem, which we are concerned with in this work, follows readily from Theorem 1 due to the fact that (deterministic) maximum k-club problem is known to be $\mathcal{NP}$-hard [9]:

Corollary 2 *The decision version of risk-averse k-club problem (RA-k) is $\mathcal{NP}$-hard, provided that risk measure ρ is proper, l.s.c., and expectation-bounded.*

The condition that risk measure ρ in the risk-averse $\mathcal{Q}$ problem (4) be l.s.c. and proper ensures that the resulting set risk function $\mathscr{R}$ is well-defined. Expectation-boundedness, on the other hand, is imposed so as to avoid situations in which the risk-averse $\mathcal{Q}$ problem becomes trivial. In the presented framework we advocate for use of coherent measures of risk when constructing the set risk function (6). It turns out, however, that if one selects $\rho(X) = \mathsf{E}X$, which is formally a coherent risk measure yet does not measure "risk", then the corresponding problem (4) is polynomially solvable, and, moreover, the solution is trivial. This can be viewed as an additional supporting argument for pursuing the *risk-averse* approach when dealing with graph-theoretical problems on graphs with stochastic vertex weights, since the traditional "expectation"-based, or risk-neutral approach to problems with stochastic vertex weights may not yield interesting results. The following proposition formalizes the above observation.

Proposition 2 *Consider the risk-averse $\mathcal{Q}$ problem* (4)*, where the risk measure ρ is such that for any $G = (V, E)$, $\mathbf{X}_G$, and $S \subseteq V$,*

$$\begin{aligned} &\arg\min\left\{\rho\left(\sum_{i\in S} u_i X_i\right) : \sum_{i\in S} u_i = 1;\ u_i \geq 0,\ \ \forall i \in S\right\} \\ &= \left\{\mathbf{u} \in \mathbb{R}^{|S|} : u_{i_S} = 1;\ u_i = 0,\ \ \forall i \in S \setminus \{i_S\}\right\}, \end{aligned} \tag{10}$$

and i_S in (10) *is computable in polynomial time. Then, the risk-averse $\mathcal{Q}$ problem is polynomially solvable, provided that property $\mathcal{Q}$ is such that one can determine in polynomial time whether there exists a $\mathcal{Q}$-subgraph of G containing a given $i \in V$.*

Proof: Obviously, condition (10) implies that

$$\mathscr{R}(S, \mathbf{X}_G) = \rho(X_{i_S}) = \min_{i\in S} \rho(X_i).$$

Then, in polynomial time one can compute $\rho(X_{i_0}) = \min_{i\in V} \rho(X_i)$ and it can be verified whether $S_0 \ni i_0$ exists such that $G[S_0]$ satisfies $\mathcal{Q}$. If not, $\rho(i_1) = \min_{i\in V\setminus\{i_0\}} \rho(X_i)$ is computed and existence of $S_1 \ni i_1$ such that $G[S_1]$ satisfies $\mathcal{Q}$ is verified in polynomial time, and so on. Clearly, the risk-averse $\mathcal{Q}$ problem can thus be solved in polynomial time. $\square$

It is easy to see that $\rho(X) = \mathsf{E}X$ constitutes a special case of the risk measure described in Proposition 2, and

$$\mathscr{R}(S, \mathbf{X}_G) = \min_{i\in S} \mathsf{E}X_i.$$

On a related note, Theorem 1 also establishes the computation complexity of risk-averse maximum hereditary subgraph problems that were discussed in our previous work [38]. Recall that property $\mathcal{Q}$ is called *hereditary with respect to induced subgraphs* if for any graph G that satisfies $\mathcal{Q}$, removal of any its vertex creates an induced subgraph that also satisfies $\mathcal{Q}$. Further, property $\mathcal{Q}$ is called *interesting* if the order of graphs that satisfy it is unbounded, and it is called *nontrivial* if it is satisfied by a single-vertex graph and is not satisfied by every graph (see, e.g., [47]).

Corollary 3 *If property $\mathcal{Q}$ is hereditary with respect to induced subgraphs, interesting, and nontrivial, and risk measure ρ is l.s.c., proper, and expectation-bounded, then the risk-averse $\mathcal{Q}$ problem is $\mathcal{NP}$-hard.*

Note that the k-club property is not hereditary with respect to induced subgraphs.

3.2 A mathematical programming formulation

In this section, we formulate the RA-k problem as a (generally nonlinear) mixed integer programming program. To this effect, let binary decision variables x_i indicate whether node $i \in V$ belongs to a subset S:

$$x_i = \begin{cases} 1, & i \in S \\ 0, & \text{otherwise.} \end{cases}$$

When the property $\mathcal{Q}$ denotes a k-club, one can choose the *edge formulation* of the maximum k-club problem proposed by Veremyev et al. [43], whereby the mathematical programming formulation of the RA-k problem takes the form

$$\min \quad \rho\Big(\sum_{i \in V} u_i \, X_i\Big) \tag{11a}$$

$$\text{s. t.} \quad \sum_{i \in V} u_i = 1, \tag{11b}$$

$$u_i \le x_i, \quad i \in V, \tag{11c}$$

$$y_{ij}^{(k)} \ge x_i + x_j - 1, \quad \forall i, j \in V, \ i \ne j, \tag{11d}$$

$$y_{ij}^{(1)} = 0, \quad \forall (i, j) \in \overline{E}, \ i \ne j, \tag{11e}$$

$$y_{ij}^{(l)} = y_{ij}^{(1)}, \quad \forall (i, j) \in E, \ l \in \{2, \dots, k\}, \tag{11f}$$

$$y_{ij}^{(l)} \le \sum_{t:(i,t) \in E} y_{tj}^{(l-1)}, \quad \forall (i, j) \in \overline{E}, \ l \in \{2, \dots, k\}, \tag{11g}$$

$$y_{ij}^{(l)} \le x_i, \ \ y_{ij}^{(l)} \le x_j, \ \ y_{ij}^{(l)} = y_{ji}^{(l)}, \quad \forall i, j \in V, \ l \in \{1, \dots, k\}, \tag{11h}$$

$$x_i \in \{0, 1\}, \ \ u_i \ge 0, \ \ y_{ij}^{(l)} \in [0, 1], \quad \forall i, j \in V, \ l \in \{1, \dots, k\}, \tag{11i}$$

where $\overline{E}$ represents the set of all complement edges of graph G. Note that nonlinearity in (11) is attributable to the possible nonlinearity of the risk measure ρ. Appropriate nonlinear mixed-integer programming solvers can be used to solve formulation (11) provided that risk measure ρ in admits a suitable mathematical programming representation. A combinatorial branch-and-bound algorithm for solving RA-k problem is described next.

3.3 A combinatorial branch-and-bound algorithm

The combinatorial branch-and-bound (BnB) algorithm for solving problem (11) processes solution space by traversing "levels" of the BnB tree to find a subgraph $G[S]$ that represents a maximal k-club of minimum risk in G as measured by (6). The algorithm begins at level $\ell = 0$ with a partial solution $Q := \emptyset$, incumbent solution $Q^* := \emptyset$, and an upper bound on risk $L^* := +\infty$ (risk induced by Q^*). Partial solution Q is composed of vertices that may potentially become a k-club during latter stages of the algorithm, while Q^* contains vertices corresponding to a maximal k-club whose risk, L^*, is the smallest up to the current stage. A set of "candidate" vertices C_ℓ is maintained at each level ℓ, from which a certain *branching* vertex v_ℓ is selected and added to the partial solution Q, or simply deleted from set C_ℓ without being added to Q. Note that the initial candidate set is $C_0 := V$. To ensure proper navigation between the levels of the BnB tree, the notation P_ℓ^+ or P_ℓ^- is used to indicate whether the last node of the BnB tree at level ℓ was created by adding v_ℓ to Q, or by deleting v_ℓ from C_ℓ without adding it to Q, respectively.

Whenever a BnB tree node is created at the consecutive level $\ell + 1$, a candidate set $C_{\ell+1}$ is constructed by removing all vertices from C_ℓ whose pairwise distances from the vertices in Q exceed k in the induced graph $G[Q \cup C_\ell]$:

$$C_{\ell+1} := \{ j \in C_\ell : d_{G[Q\cup C_\ell]}(i, j) \le k, \ \forall i \in Q \}.$$

Observe that the refinement of C_ℓ may disrupt the structural integrity of the partial solution if the eliminated candidate vertices serve as distance intermediaries (i.e., comprise the shortest paths) between the vertices in Q. In other words, the distance between at least one pair of vertices $i, j \in Q$ exceeds k upon removal of one or more vertices from C_ℓ when constructing $C_{\ell+1}$. Due to this inherent distance-based dependence of k-clubs, additional considerations are warranted whenever creating a BnB node by either adding or deleting a vertex v_ℓ (i.e., P_ℓ^+ or P_ℓ^-, respectively). Therefore, the necessary structural properties of Q and $C_{\ell+1}$ at each BnB node are

(C1) Q is a k-clique in $G[Q \cup C_{\ell+1}]$, and

(C2) $d_{G[Q\cup C_\ell]}(i, j) \le k, \ \ \forall i \in Q, \ \ \forall j \in C_{\ell+1}$.

After constructing set $C_{\ell+1}$ (condition (C2) is satisfied by definition of $C_{\ell+1}$), if vertices in $C_\ell \setminus C_{\ell+1}$ do serve as distance intermediaries, their removal imposes violations with respect to condition (C1). In such cases, Q cannot become a k-club by exploring deeper levels of the tree and the corresponding node of the BnB tree is fathomed[2] by infeasibility via violation of condition (C1).

Whenever condition (C1) is satisfied, the next step entails evaluating the quality of the solution that can be obtained from the subgraph induced by vertices in $Q \cup C_{\ell+1}$. An exact approach for directly finding a k-club with the lowest possible risk contained in $G[Q \cup C_{\ell+1}]$ would involve solving problem (11) with $x_i = 0$ for all $i \in V \setminus (Q \cup C_{\ell+1})$; we denote the corresponding solution by $\mathcal{S}(Q \cup C_{\ell+1}; \mathbf{X}_G)$. Solving such a (nonlinear) mixed 0–1 program at every node of the BnB tree is clearly impractical. Instead, the following relaxation problem is utilized to obtain a valid lower bound on $\mathcal{S}(Q \cup C_{\ell+1}; \mathbf{X}_G)$:

$$\begin{aligned} \mathcal{L}(Q \cup C_{\ell+1}; \mathbf{X}_G) := \min \quad & \rho\Big(\sum_{i\in V} u_i \, X_i\Big) \\ \text{s. t.} \quad & \sum_{i\in V} u_i = 1, \\ & u_i = 0, \quad i \in V \setminus (Q \cup C_{\ell+1}) \\ & u_i \ge 0, \quad i \in Q \cup C_{\ell+1}. \end{aligned} \tag{12}$$

If $\mathcal{L}(Q \cup C_{\ell+1}; \mathbf{X}_G) \ge L^*$, then the corresponding node of the BnB tree is fathomed by bound due to the fact that sequential refinement can not achieve a further reduction in risk.

In the case when $\mathcal{L}(Q \cup C_{\ell+1}; \mathbf{X}_G) < L^*$ and $Q \cup C_{l+1}$ is a k-club, the new incumbent solution will be $Q^* = Q \cup C_{l+1}$ and the global upper bound on risk is updated, $L^* = \mathcal{L}(Q \cup C_{\ell+1}; \mathbf{X}_G)$. In

[2] Indicated in Algorithm 1 by the assignment "fathom := True".

this case, the current BnB node is fathomed by feasibility. If, however, $\mathcal{L}(Q \cup C_{\ell+1}; \mathbf{X}_G) < L^*$ and $Q \cup C_{l+1}$ is not a k-club, a branching vertex $v_{\ell+1}$ is selected at the next level $\ell + 1$ and BnB node $P^+_{\ell+1}$ will be processed.

After fathoming a BnB node, the algorithm backtracks as follows. If the current BnB node is of type P^+_ℓ, then the vertex v_ℓ is removed from Q, and the node associated with the deletion of v_ℓ, P^-_ℓ, is created. On the other hand, if the BnB node is of type P^-_ℓ, the algorithm sequentially backtracks to the last level, $\ell' < \ell$, associated with a node of type $P^+_{\ell'}$. The node $P^-_{\ell'}$ is then constructed by removing the branching vertex $v_{\ell'}$ from Q. Observe that a node can only be of form P^-_ℓ, after P^+_ℓ has been fathomed/processed.

Empirical observations suggest that branching on a vertex v_ℓ with the smallest value of $\rho(X_{v_\ell})$ or $\mathsf{E}X_{v_\ell}$ can significantly enhance computational performance. To this end, the vertices in any candidate set C_ℓ are ordered in descending order with respect to their risks $\rho(X_i)$ or expected values $\mathsf{E}X_i$, and the last vertex in C_ℓ is always selected when adding vertex v_ℓ to the partial solution Q. The described branch-and-bound algorithm procedure for RA-k problem is formalized in Algorithm 1.

As shown in [17], it is important to mention that the number of leaf nodes in the BnB search tree of Algorithm 1 is $O^*(1.62^{|V|})$, where the modified notation "$O^*\big(g(|V|)\big)$" implies $O\big(g(|V|) \cdot poly(|V|)\big)$ for some polynomial function $poly(|V|)$. Additionally, at each node of the search tree, all pair distances can be computed in $O(|V|^3)$ time and we solve a linear program to obtain a lower bound on the optimal solution of the subtree rooted at that node. Therefore, Algorithm 1 runs in $O^*(1.62^{|V|})$.

Algorithm 1: Combinatorial branch-and-bound algorithm

```
Initialize: ℓ := 0; C_0 := V; Q := ∅; Q* := ∅; L* = ∞; node := P_0^+; fathom := False;
while ℓ ≥ 0 do
    if node = P_ℓ^+ then
        select a vertex v_ℓ ∈ C_ℓ;
        C_ℓ := C_ℓ \ {v_ℓ};
        Q := Q ∪ {v_ℓ};
    else
        Q := Q \ {v_ℓ};
    C_{ℓ+1} := {j ∈ C_ℓ : d_{G[Q∪C_ℓ]}(i, j) ≤ k, ∀i ∈ Q};
    if Q is a k-clique in G[Q ∪ C_{ℓ+1}] then
        if 𝓛(Q ∪ C_{ℓ+1}) < L* then
            if Q ∪ C_{ℓ+1} is a k-club then
                Q* := Q ∪ C_{ℓ+1};
                L* := 𝓛(Q ∪ C_{ℓ+1});
                fathom := True;
        else
            fathom := True;
    else
        fathom := True;
    if fathom = True then
        while ℓ ≥ 0 and node = P_ℓ^- do
            ℓ := ℓ − 1;
        node := P_ℓ^-;
        fathom := False;
    else
        ℓ := ℓ + 1;
        node := P_ℓ^+;
return Q*;
```

4 Case study: Risk-averse k-club problem with higher moment coherent risk measures

In this section, we present a computational framework for problem (11) and conduct numerical experiments demonstrating the computational performance of the proposed BnB algorithm. To this end, we adopt higher moment coherent risk measures to quantify the risk as described next.

4.1 Higher moment coherent risk measures

The class of higher-moment coherent risk (HMCR) measures was introduced in [27] as optimal values to the following stochastic programming problem:

$$\mathrm{HMCR}_{\alpha,p}(X) = \min_{\eta\in\mathbb{R}} \; \eta + (1-\alpha)^{-1} \left\| (X-\eta)^+ \right\|_p, \quad \alpha \in (0,1), \quad p \geq 1, \tag{13}$$

where $X^+ = \max\{0, X\}$ and $\|X\|_p = \left(\mathsf{E}|X|^p\right)^{1/p}$. Mathematical programming problems that contain HMCR measures can be formulated using p-order cone constraints. Typically, in stochastic programming models, the set of random events Ω is assumed to be discrete, $\Omega = \{\omega_1, \ldots, \omega_N\}$, with the probabilities $\mathsf{P}\{\omega_k\} = \pi_k > 0$, and $\pi_1 + \cdots + \pi_N = 1$. The corresponding mathematical programming model (11) with $\rho(X) = \mathrm{HMCR}_{p,\alpha}(X)$ takes the following mixed 0–1 p-order cone programming form:

$$
\begin{aligned}
\min \quad & \eta + (1-\alpha)^{-1} t_0 \\
\text{s. t.} \quad & t_0 \geq \|(t_1, \ldots, t_N)\|_p, \\
& \pi_k^{-1/p} y_k \geq \sum_{i \in V} u_i X_{ik} - \eta, \quad k = 1, \ldots, N, \\
& \sum_{i \in V} u_i = 1, \\
& u_i \leq x_i, \ \ i \in V, \\
& \text{(11d)} - \text{(11i)}, \\
& t_k \geq 0, \ k = 0, \ldots, N,
\end{aligned}
\tag{14}
$$

where X_{ik} represents the realization of the stochastic weight of vertex $i \in V$ under scenario $k \in \{1, \ldots, N\}$. Analogously, the lower bound problem (12) takes the form

$$
\begin{aligned}
\mathcal{L}(Q \cup C_{\ell+1}; \mathbf{X}_G) = \min \quad & \eta + (1-\alpha)^{-1} t_0 \\
\text{s. t.} \quad & t_0 \geq \|t_1, \ldots, t_N\|_p, \\
& \pi_k^{-1/p} t_k \geq \sum_{i \in V} u_i X_{ik} - \eta, \quad k = 1, \ldots, N, \\
& \sum_{i \in V} u_i = 1, \\
& u_i \geq 0, \ \ i \in Q \cup C_{\ell+1}, \\
& u_i = 0, \ \ i \in V \setminus (Q \cup C_{\ell+1}), \\
& t_k \geq 0, \ k = 0, \ldots, N.
\end{aligned}
\tag{15}
$$

For instances when $p \in \{1, 2\}$, problems (14) and (15) reduce to linear programming (LP) and second order cone programming (SOCP) models, respectively. However, in cases when when $p \in (1, 2) \cup (2, \infty)$ the p-cone is not self-dual and there exist no efficient long-step self-dual interior point solution methods. Consequently, we employ solution methods for p-order cone programming problems that are based on polyhedral approximations of p-order cones [45] and representation of rational-order p-cones via second order cones [31].

4.2 Setup of the numerical experiments and results

Numerical experiments of the risk-averse k-club problem for $k = 2, 3, 4$ were conducted on randomly generated Erdös-Rényi graphs of orders $|V| = 50, 100, 200$ with average densities $D(G) = 0.0125$, 0.025, 0.05, 0.1, and 0.15. The specified densities were chosen due to empirical observations indicating that a graph of order $|V| \geq 50$ commonly reduces to a 2-club when the density is in the range [0.15, 0.25]. Clearly, this effect is even more pronounced for $k > 2$. The stochastic weights of graphs' vertices were generated as i.i.d. samples from the uniform $U(0, 1)$ distribution. Scenario sets with $N = 250$ scenarios were generated for each combination of graph order and density. The HMCR risk measures (13) with $p = 1, 2, 3$, and $\alpha = 0.9$ were used.

The BnB algorithm has been coded in C++, and we used the CPLEX Simplex and Barrier solvers for the polyhedral approximations and SOCP reformulations of the p-order cone programming lower bound problem (15), respectively (see [25]). For instances when $p = 1$, the CPLEX Simplex solver was utilized to solve problem (15) directly. The computations were conducted on an Intel Xeon 3.30GHz PC with 128GB RAM, and CPLEX 12.6 solver in Windows 7 64-bit environment was used.

The computational performance of the mathematical programming model (14) was compared with that of developed BnB algorithm. In the case of $p = 1$, problem (14) was solved with CPLEX Mixed Integer Programming (MIP) solver. The CPLEX MIP Barrier solver was used for the SOCP version in the case of $p = 2$, and using the SOCP reformulation in the case of $p = 3$.

Tables 1– 3 present the computational times and the best objective values averaged over five instances for each graph configuration, as well as the number of instances for which an optimal solution was attained within a 3600 second time limit. The reported average time is calculated by only considering the instances where the problem was solved to optimality within the time limit, while the reported average objective value is calculated by only considering the instances in which at least a feasible solution is found within the time limit. The symbol "—" was used to indicate that the time limit was exceeded, and cells containing "NA" correspond to instances for which solution process failed due to CPLEX running out of memory. Table 1 demonstrates that the BnB algorithm significantly outperforms the CPLEX MIP solver over all the listed graph configurations when $k = 2$, achieving up to an order of magnitude of improvement in computational time. Further, observe that the quality of the average best objectives obtained by the BnB algorithm was superior whenever both methods failed to reach an optimal solution within the time limit. In cases when CPLEX failed due to memory capacity issues, the BnB algorithm either attained an optimal solution or an incumbent solution, in which case the average solution associated with the best incumbent solutions are provided. Note that the performance of both algorithms decreases for higher values of p. This becomes particularly pronounced for $p = 3$ and $|V| = 200$ in Table 1, where CPLEX could not manage any of the corresponding instances due to the increased problem size associated with the cutting-plane algorithm for solving polyhedral approximations of p-order cone programming problems, while the BnB algorithm only solved eleven instances within the time limit.

A similar improvement in performance can be observed for $k = 3$ and $k = 4$ in Tables 2–3. As k increases, the number of time limit and memory capacity limit violations for CPLEX increases, further demonstrating the applicability of the proposed BnB method. This observable disadvantage associate with model (11) results from the fact that the number of constraints in model (11) rapidly increases with k, thus overwhelming the solver in many cases. All the instances in Table 3 with $|V| = 200$ are of this type.

Based on the results presented in Tables 1–3, it is worth noting that as $D(G)$ increases for a given p and $|V|$, the average computation time for the BnB algorithm increases, reaches a maximum value, and then decreases. This is due to the fact that once $D(G)$ is large enough, graph G tends to contain larger components of lower diameter that can be detected at the early stages of the BnB algorithm. Another interesting observation is that for a given p and $D(G)$, if $|V|$ is large enough, the average computation time for BnB algorithm decreases as $|V|$ increases. For instance, in Table 2, for $p = 2$ and $D(G) = 0.1$, none of the instances with $|V| = 100$ were solved to optimality, while all the instances with $|V| = 200$ were solved to optimality within 4.05 seconds on average. This observation can be justified by the fact that for a given expected edge density $D(G)$, if $|V|$ is sufficiently large, the diameter of the random graph decreases as $|V|$ increases (see, e.g., [11], p. 62). Therefore, in these cases, the graphs with larger $|V|$

tend to have larger components of low diameter that can likewise be detected during the early stages of the BnB algorithm.

In order to demonstrate the applicability of our algorithms on real-life graphs, Tables 4- 6 present the results obtained from solving various DIMACS graph instances with the same number of scenarios and distribution of uncertain vertex weights as above. Observe that the BnB method outperforms CPLEX over the vast majority of tested instances, and more than two orders of magnitude in improvements were observed for various cases. However, in several cases even the BnB algorithm failed to obtain an incumbent solution within the time limit (denoted by "∞"), underscoring the complex nature of many real-life graphs.

			$p = 1$			$p = 2$			$p = 3$		
				$\|V\|$			$\|V\|$			$\|V\|$	
$D(G)$	Algorithm	Output	50	100	200	50	100	200	50	100	200
0.0125	CPLEX	Time (s)	0.95	4.46	61.64	32.18	91.33	3043.69	129.07	278.05	NA
		Instance	5	5	5	5	5	1	5	5	0
		Objective	0.23	0.21	0.19	0.28	0.25	0.37	0.30	0.25	NA
	BnB	Time (s)	0.23	1.04	5.01	8.63	25.13	80.27	35.64	100.08	301.68
		Instance	5	5	5	5	5	5	5	5	5
		Objective	0.23	0.21	0.19	0.28	0.25	0.21	0.30	0.25	0.21
0.025	CPLEX	Time (s)	1.44	7.49	177.34	46.96	233.35	—	93.66	352.55	NA
		Instance	5	5	5	5	5	0	5	5	0
		Objective	0.23	0.20	0.17	0.28	0.23	0.54	0.29	0.23	NA
	BnB	Time (s)	0.24	1.11	7.62	10.90	30.35	167.98	51.66	141.57	746.82
		Instance	5	5	5	5	5	5	5	5	5
		Objective	0.23	0.20	0.17	0.28	0.23	0.19	0.29	0.23	0.19
0.05	CPLEX	Time (s)	1.92	14.64	2185.63	60.53	472.10	—	123.17	776.51	NA
		Instance	5	5	5	5	5	0	5	5	0
		Objective	0.20	0.18	0.15	0.23	0.19	0.20	0.24	0.19	NA
	BnB	Time (s)	0.26	2.02	37.88	15.43	75.50	1087.92	65.07	368.43	3051.66
		Instance	5	5	5	5	5	5	5	5	1
		Objective	0.20	0.18	0.15	0.23	0.19	0.16	0.24	0.19	0.16
0.1	CPLEX	Time (s)	4.25	322.01	—	150.96	—	—	423.76	—	NA
		Instance	5	5	0	5	0	0	5	0	0
		Objective	0.18	0.15	0.14	0.20	0.18	0.41	0.20	0.18	NA
	BnB	Time (s)	0.59	28.51	—	38.03	1451.43	—	183.78	—	—
		Instance	5	5	0	5	5	0	5	0	0
		Objective	0.18	0.15	0.14	0.20	0.16	0.15	0.20	0.16	0.15
0.15	CPLEX	Time (s)	9.48	2832.38	—	1055.83	—	—	1862.11	—	NA
		Instance	5	2	0	5	0	0	5	0	0
		Objective	0.17	0.14	0.18	0.18	0.16	0.17	0.18	0.16	NA
	BnB	Time (s)	2.41	2033.67	—	164.06	—	—	707.16	—	—
		Instance	5	3	0	5	0	0	5	0	0
		Objective	0.17	0.14	0.18	0.18	0.14	0.20	0.18	0.15	0.22

Table 1: Average computation times (in seconds), number of instances solved to optimality (out of five) and the average best objective values obtained by solving problem (11) using the proposed BnB algorithm and CPLEX with $k = 2$ and risk measure (13).

			$p=1$			$p=2$			$p=3$		
			$\|V\|$			$\|V\|$			$\|V\|$		
$D(G)$	Algorithm	Output	50	100	200	50	100	200	50	100	200
0.0125	CPLEX	Time (s)	0.88	6.12	NA	14.16	148.36	NA	86.09	258.75	NA
		Instance	5	5	0	5	5	0	5	5	0
		Objective	0.22	0.19	NA	0.27	0.22	NA	0.27	0.21	NA
	BnB	Time (s)	0.23	1.04	6.80	8.84	29.01	162.51	36.64	113.63	708.45
		Instance	5	5	5	5	5	5	5	5	5
		Objective	0.22	0.19	0.16	0.27	0.22	0.18	0.27	0.21	0.18
0.025	CPLEX	Time (s)	1.26	12.68	NA	27.07	516.78	NA	69.69	577.47	NA
		Instance	5	5	0	5	5	0	5	5	0
		Objective	0.21	0.18	NA	0.24	0.19	NA	0.25	0.19	NA
	BnB	Time (s)	0.24	1.59	81.50	11.28	65.54	2075.28	52.72	286.93	—
		Instance	5	5	5	5	5	4	5	5	0
		Objective	0.21	0.18	0.15	0.24	0.19	0.15	0.25	0.19	0.15
0.05	CPLEX	Time (s)	2.79	287.23	NA	163.45	—	NA	385.90	—	NA
		Instance	5	5	0	5	0	0	5	0	0
		Objective	0.17	0.14	NA	0.19	0.17	NA	0.19	0.16	NA
	BnB	Time (s)	0.43	44.13	—	29.41	1060.88	—	131.34	1531.64	—
		Instance	5	5	0	5	4	0	5	1	0
		Objective	0.17	0.14	0.14	0.19	0.15	0.18	0.19	0.15	0.17
0.1	CPLEX	Time (s)	14.27	25.41	NA	2656.62	3425.15	NA	2797.45	3311.62	NA
		Instance	5	2	0	5	1	0	2	1	0
		Objective	0.15	0.11	NA	0.15	0.11	NA	0.15	0.11	NA
	BnB	Time (s)	3.00	719.53	3.70	367.80	—	4.05	941.06	—	4.96
		Instance	5	3	5	5	0	5	5	0	5
		Objective	0.15	0.11	0.10	0.15	0.11	0.10	0.15	0.11	0.10
0.15	CPLEX	Time (s)	50.30	480.23	NA	2329.51	—	NA	1003.57	—	NA
		Instance	5	5	0	2	0	0	3	0	0
		Objective	0.13	0.11	NA	0.13	0.12	NA	0.13	0.12	NA
	BnB	Time (s)	2.50	0.29	4.63	762.31	0.50	4.94	998.11	1.51	5.89
		Instance	5	5	5	5	5	5	4	5	5
		Objective	0.13	0.11	0.10	0.13	0.11	0.10	0.13	0.11	0.10

Table 2: Average computation times (in seconds), number of instances solved to optimality (out of five) and the average best objective values obtained by solving problem (11) using the proposed BnB algorithm and CPLEX with $k = 3$ and risk measure (13).

5 Conclusions

We have considered an RA-k problem which entails finding a k-club of minimum risk in a graph. HMCR risk measures were utilized for quantifying the distributional information of the stochastic factors associated with vertex weights. It was shown that the decision version of RA-k problem is $\mathcal{NP}$-hard for any fixed positive integer k, and the optimal solutions are maximal k-clubs. A combinatorial BnB solution algorithm was developed and tested on a special case of RA-k problem when $k = 2, 3, 4$. Numerical experiments on randomly generated graphs of various configurations suggest that the proposed BnB algorithm can significantly reduce solution times in comparison with the mathematical programming model solved using CPLEX MIP solver.

			$p = 1$			$p = 2$			$p = 3$		
			$\|V\|$			$\|V\|$			$\|V\|$		
$D(G)$	Algorithm	Output	50	100	200	50	100	200	50	100	200
0.0125	CPLEX	Time (s)	0.90	7.30	NA	27.09	206.30	NA	118.25	299.46	NA
		Instance	5	5	0	5	5	0	5	5	0
		Objective	0.21	0.17	NA	0.25	0.18	NA	0.26	0.19	NA
	BnB	Time (s)	0.23	1.13	15.13	8.25	31.98	409.93	47.41	160.66	2001.38
		Instance	5	5	5	5	5	5	5	5	5
		Objective	0.21	0.17	0.14	0.25	0.18	0.15	0.26	0.19	0.15
0.025	CPLEX	Time (s)	1.60	21.97	NA	50.10	1475.67	NA	79.91	1714.83	NA
		Instance	5	5	0	5	5	0	5	5	0
		Objective	0.19	0.15	NA	0.22	0.16	NA	0.23	0.16	NA
	BnB	Time (s)	0.23	2.46	—	11.58	91.57	—	63.31	514.33	—
		Instance	5	5	0	5	5	0	5	5	0
		Objective	0.19	0.15	0.12	0.22	0.16	0.13	0.23	0.16	0.13
0.05	CPLEX	Time (s)	4.34	—	NA	461.18	—	NA	929.33	—	NA
		Instance	5	0	0	5	0	0	5	0	0
		Objective	0.16	0.12	NA	0.16	0.12	NA	0.16	0.12	NA
	BnB	Time (s)	0.66	728.07	2.71	35.37	—	3.06	177.83	—	4.23
		Instance	5	5	5	5	0	5	5	0	5
		Objective	0.16	0.12	0.10	0.16	0.12	0.10	0.16	0.12	0.10
0.1	CPLEX	Time (s)	33.72	1776.46	NA	898.37	449.72	NA	493.52	500.88	NA
		Instance	5	4	0	3	3	0	4	3	0
		Objective	0.13	0.13	NA	0.13	0.11	NA	0.13	0.11	NA
	BnB	Time (s)	4.63	0.25	3.71	187.73	0.47	4.07	236.05	1.72	5.33
		Instance	5	5	5	5	5	5	5	5	5
		Objective	0.13	0.11	0.10	0.13	0.11	0.10	0.13	0.11	0.10
0.15	CPLEX	Time (s)	25.40	2503.89	NA	282.75	—	NA	271.86	—	NA
		Instance	5	5	0	5	0	0	5	0	0
		Objective	0.13	0.11	NA	0.13	0.12	NA	0.13	0.12	NA
	BnB	Time (s)	0.04	0.30	4.63	0.22	0.53	4.97	1.54	1.95	6.42
		Instance	5	5	5	5	5	5	5	5	5
		Objective	0.13	0.11	0.10	0.13	0.11	0.10	0.13	0.11	0.10

Table 3: Average computation times (in seconds), number of instances solved to optimality (out of five) and the average best objective values obtained by solving problem (11) using the proposed BnB algorithm and CPLEX with $k = 4$ and risk measure (13).

6 Acknowledgements

This research was performed while the first author held a National Research Council Research Associateship Award at the Air Force Research Laboratory. This work was supported in part by the AFOSR grant FA9550-12-1-0142, DTRA grant HDTRA1-14-1-0065, and the U.S. Department of Air Force grant FA8651-14-2-0003. In addition, support by the AFRL Mathematical Modeling and Optimization Institute is gratefully acknowledged.

References

[1] J. Abello, P. Pardalos, and M. Resende. On maximum clique problems in very large graphs. In J. Abello and J. Vitter, editors, *External memory algorithms and visualization*, volume 50 of *DI-*

				$p = 1$		$p = 2$		$p = 3$	
Graph Name	$\|V\|$	$\|E\|$	Output	CPEX	BnB	CPLEX	BnB	CPLEX	BnB
adjnoun.clq	112	425	Time (s) Objective	22.25 0.20	4.95 0.20	— ∞	92.12 0.21	— ∞	368.83 0.21
celegans metabolic.clq	453	2025	Time (s) Objective	3404.14 0.10	55.40 0.10	— ∞	191.94 0.11	— ∞	292.25 0.11
celegansneural.clq	297	2148	Time (s) Objective	585.59 0.10	— 0.10	— ∞	— 0.10	— ∞	— 0.10
chesapeake.clq	39	170	Time (s) Objective	1.33 0.23	0.11 0.23	129.44 0.24	4.85 0.24	106.52 0.24	19.50 0.24
dolphins.clq	62	159	Time (s) Objective	4.90 0.35	0.89 0.35	193.18 0.41	30.05 0.41	402.20 0.40	161.37 0.40
email.clq	1133	5451	Time (s) Objective	— ∞	1221.39 0.19	NA NA	— 0.19	NA NA	— 0.21
football.clq	115	613	Time (s) Objective	179.92 0.33	9.21 0.33	— 0.37	314.28 0.36	— NA	1575.35 0.36
jazz.clq	198	2742	Time (s) Objective	— ∞	— ∞	— ∞	— ∞	— ∞	— ∞
karate.clq	34	78	Time (s) Objective	1.83 0.32	0.14 0.32	60.69 0.35	8.22 0.35	76.15 0.35	41.90 0.35
lesmis.clq	77	254	Time (s) Objective	7.35 0.20	8.19 0.20	— ∞	— 0.22	— ∞	— 0.22
netscience.clq	1589	2742	Time (s) Objective	— ∞	479.99 0.23	NA NA	3380.12 0.24	NA NA	— 0.24
polblogs.clq	1490	16715	Time (s) Objective	— ∞	— 0.16	NA NA	— 0.16	NA NA	— 0.16
polbooks.clq	105	441	Time (s) Objective	22.42 0.26	4.03 0.26	— ∞	62.73 0.27	— ∞	303.97 0.27

Table 4: Computation times (in seconds) and the best objective values obtained by solving problem (11) for various DIMACS graph instances using the proposed BnB algorithm and CPLEX with $k = 2$ and risk measure (13).

MACS Series on Discrete Mathematics and Theoretical Computer Science, pages 119–130. American Mathematical Society, 1999.

[2] J. Abello, M. Resende, and S. Sudarsky. Massive quasi-clique detection. In S. Rajsbaum, editor, *LATIN 2002: Theoretical Informatics*, pages 598–612, London, 2002. Springer-Verlag.

[3] R. D. Alba. A graph-theoretic definition of a sociometric clique. *Journal of Mathematical Sociology*, 3:3–113, 1973.

[4] Y. P. Aneja, R. Chandrasekaran, and K. P. K. Nair. Maximizing residual flow under an arc destruction. *Networks*, 38(4):194–198, 2001.

[5] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath. Coherent measures of risk. *Mathematical Finance*, 9(3):203–228, 1999.

[6] A. Atamtrk and M. Zhang. Two-stage robust network flow and design under demand uncertainty. *Operations Research*, 55(4):662–673, 2007.

				$p = 1$		$p = 2$		$p = 3$	
DIMACS	$\|V\|$	$\|E\|$	Output	CPEX	BnB	CPLEX	BnB	CPLEX	BnB
adjnoun.clq	112	425	Time (s) Objective	554.92 0.17	2911.88 0.17	— ∞	— 0.17	— ∞	— 0.17
celegans metabolic.clq	453	2025	Time (s) Objective	— ∞	— 0.10	NA NA	— 0.11	NA NA	— 0.11
celegansneural.clq	297	2148	Time (s) Objective	— ∞	20.81 0.10	NA NA	21.19 0.10	NA NA	21.81 0.10
chesapeake.clq	39	170	Time (s) Objective	1.23 0.22	0.03 0.22	67.80 0.23	0.39 0.23	74.99 0.23	0.78 0.23
dolphins.clq	62	159	Time (s) Objective	16.24 0.26	1.25 0.26	2644.07 0.27	59.13 0.27	— ∞	221.67 0.27
email.clq	1133	5451	Time (s) Objective	— ∞	— 0.14	NA NA	— 0.14	NA NA	— 0.14
football.clq	115	613	Time (s) Objective	— ∞	2935.65 0.19	— ∞	— 0.20	— ∞	— 0.21
jazz.clq	198	2742	Time (s) Objective	— ∞	— 0.12	— ∞	— 0.12	— ∞	— 0.12
karate.clq	34	78	Time (s) Objective	1.42 0.28	0.27 0.28	— ∞	20.56 0.29	— ∞	68.19 0.29
lesmis.clq	77	254	Time (s) Objective	5.87 0.19	0.12 0.19	750.07 0.19	0.30 0.19	520.12 0.19	1.08 0.19
netscience.clq	1589	2742	Time (s) Objective	— ∞	554.77 0.20	NA NA	— 0.20	NA NA	— 0.20
polblogs.clq	1490	16715	Time (s) Objective	— ∞	— ∞	NA NA	— ∞	NA NA	— ∞
polbooks.clq	105	441	Time (s) Objective	202.53 0.21	30.70 0.21	— ∞	901.68 0.22	— ∞	3468.82 0.22

Table 5: Computation times (in seconds) and the best objective values obtained by solving problem (11) for various DIMACS graph instances using the proposed BnB algorithm and CPLEX with $k = 3$ and risk measure (13).

[7] L. Babel. A fast algorithm for the maximum weight clique problem. *Computing*, 52(1):31–38, 1994.

[8] E. Balas and C. S. Yu. Finding a maximum clique in an arbitrary graph. *SIAM J. Comput.*, 15(4):1054–1068, Nov. 1986.

[9] B. Balasundaram, S. Butenko, and S. Trukhanov. Novel approaches for analyzing biological networks. *Journal of Combinatorial Optimization*, 10(1):23–39, 2005.

[10] B. Balasundaram and F. M. Pajouh. Graph theoretic clique relaxations and applications. In P. M. Pardalos, D.-Z. Du, and R. Graham, editors, *Handbook of Combinatorial Optimization*, pages 1559–1598. Springer, 2nd edition, 2013.

[11] A. Barabasi. *Network Science*. Center for Complex Network Research at Northeastern University (http://barabasilab.neu.edu/networksciencebook/downlPDF.html), Boston, MA, 2012.

				$p=1$		$p=2$		$p=3$	
DIMACS	$\lvert V\rvert$	$\lvert E\rvert$	Output	CPEX	BnB	CPLEX	BnB	CPLEX	BnB
adjnoun.clq	112	425	Time (s) Objective	— ∞	18.57 0.15	— ∞	282.90 0.15	— ∞	449.63 0.15
celegans metabolic.clq	453	2025	Time (s) Objective	— ∞	1953.44 0.11	NA NA	2081.61 0.12	NA NA	2123.89 0.12
celegansneural.clq	297	2148	Time (s) Objective	— ∞	20.73 0.10	NA NA	21.20 0.10	NA NA	21.67 0.10
chesapeake.clq	39	170	Time (s) Objective	1.70 0.22	0.03 0.22	95.35 0.23	0.23 0.23	84.43 0.23	0.72 0.23
dolphins.clq	62	159	Time (s) Objective	37.82 0.24	13.21 0.24	— ∞	542.94 0.24	— ∞	2640.58 0.24
email.clq	1133	5451	Time (s) Objective	NA NA	— ∞	NA NA	— ∞	NA NA	— ∞
football.clq	115	613	Time (s) Objective	326.48 0.15	0.42 0.15	435.09 0.15	0.87 0.15	397.31 0.15	1.45 0.15
jazz.clq	198	2742	Time (s) Objective	— ∞	— 0.12	NA NA	— 0.12	NA NA	— 0.12
karate.clq	34	78	Time (s) Objective	1.89 0.23	0.03 0.23	51.11 0.24	0.78 0.24	57.43 0.24	3.82 0.24
lesmis.clq	77	254	Time (s) Objective	7.86 0.19	0.14 0.19	948.37 0.19	0.38 0.19	1067.50 0.19	1.17 0.19
netscience.clq	1589	2742	Time (s) Objective	NA NA	— 0.19	NA NA	— 0.16	NA NA	— 0.18
polblogs.clq	1490	16715	Time (s) Objective	NA NA	— ∞	NA NA	— ∞	NA NA	— ∞
polbooks.clq	105	441	Time (s) Objective	786.70 0.18	— 0.19	— ∞	— 0.20	— ∞	— 0.20

Table 6: Computation times (in seconds) and the best objective values obtained by solving problem (11) for various DIMACS graph instances using the proposed BnB algorithm and CPLEX with $k = 4$ and risk measure (13).

[12] V. Boginski, S. Butenko, and P. Pardalos. Mining market data: a network approach. *Computers & Operations Research*, 33(11):3171–3184, 2006.

[13] J.-M. Bourjolly, G. Laporte, and G. Pesant. An exact algorithm for the maximum k-club problem in an undirected graph. *European Journal of Operational Research*, 138(1):21 – 28, 2002.

[14] A. M. Campbell and B. W. Thomas. Probabilistic traveling salesman problem with deadlines. *Transportation Science*, 42(1):1–21, 2008.

[15] R. Carmo and A. Zge. Branch and bound algorithms for the maximum clique problem under a unified framework. *Journal of the Brazilian Computer Society*, 18(2):137–151, 2012.

[16] R. Carraghan and P. M. Pardalos. An exact algorithm for the maximum clique problem. *Operations Research Letters*, 9(6):375 – 382, 1990.

[17] M.-S. Chang, L.-J. Hung, C.-R. Lin, and P.-C. Su. Finding large k-clubs in undirected graphs. *Computing*, 95(9):739–758, 2013.

[18] D. J. Cook and L. B. Holder. Graph-based data mining. *IEEE Intelligent Systems*, 15(2):32–41, 2000.

[19] F. Delbaen. Coherent risk measures on general probability spaces. pages 1–37, 2002.

[20] G. D. Glockner and G. L. Nemhauser. A dynamic network flow problem with uncertain arc capacities: Formulation and problem structure. *Operations Research*, 48(2):233–242, 2000.

[21] A. Gupta, V. Nagarajan, and R. Ravi. Technical noteapproximation algorithms for vrp with stochastic demands. *Operations Research*, 60(1):123–127, 2012.

[22] S. Hill, F. Provost, and C. Volinsky. Network-based marketing: Identifying likely adopters via consumer networks. *Statistical Science*, 22:256–275, 2006.

[23] D. Iacobucci and N. Hopkins. Modeling dyadic interactions and networks in marketing. *Journal of Marketing Research*, 24:5–17, 1992.

[24] J. Konc and D. Janezic. An improved branch and bound algorithm for the maximum clique problem. *proteins*, 4:5, 2007.

[25] P. Krokhmal and P. Soberanis. Risk optimization with p-order conic constraints: A linear programming approach. *European Journal of Operational Research*, 301(3):653–671, 2010.

[26] P. Krokhmal, M. Zabarankin, and S. Uryasev. Modeling and optimization of risk. *Surveys in Operations Researh and Management Science*, 16(2):49–66, 2011.

[27] P. A. Krokhmal. Higher moment coherent risk measures. *Quantitative Finance*, 7:373–387, 2007.

[28] D. Kumlander. A new exact algorithm for the maximum-weight clique problem based on a heuristic vertex-coloring and a backtrack search. In *Proceedings of the Fourth International Conference on Engineering Computational Technology*, pages 137–138. Civil-Comp Press, 2004.

[29] R. Luce. Connectivity and generalized cliques in sociometric group structure. *Psychometrika*, 15(2):169–190, 1950.

[30] R. Mokken. Cliques, clubs and clans. *Quality and Quantity*, 13(2):161–173, 1979.

[31] Y. Morenko, A. Vinel, Z. Yu, and P. Krokhmal. On p-cone linear discrimination. *European Journal of Operational Research*, 231(3):784–789, 2013.

[32] P. R. J. Östergård. A new algorithm for the maximum-weight clique problem. *Nordic J. of Computing*, 8(4):424–436, Dec. 2001.

[33] P. R. J. Östergård. A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics*, 120(1–3):197–207, 2002. Special Issue devoted to the 6th Twente Workshop on Graphs and Combinatorial Optimization.

[34] F. M. Pajouh and B. Balasundaram. On inclusionwise maximal and maximum cardinality k-clubs in graphs. *Discrete Optimization*, 9(2):84 – 97, 2012.

[35] J. Pattillo, A. Veremyev, S. Butenko, and V. Boginski. On the maximum quasi-clique problem. *Discrete Applied Mathematics*, 161(1–2):244–257, 2013.

[36] R. T. Rockafellar and S. Uryasev. The fundamental risk quadrangle in risk management, optimization and statistical estimation. *Surveys in Operations Research and Management Science*, 18:33–53, 2013.

[37] R. T. Rockafellar, S. Uryasev, and M. Zabarankin. Generalized deviations in risk analysis. *Finance and Stochastics*, 10(1):51–74, 2006.

[38] M. Rysz, M. Mirghorbani, P. Krokhmal, and E. Pasiliao. On risk-averse maximum weighted subgraph problems. *Journal of Combinatorial Optimization*, 28(1):167–185, 2014.

[39] A. Schfer, C. Komusiewicz, H. Moser, and R. Niedermeier. Parameterized computational complexity of finding small-diameter subgraphs. *Optimization Letters*, 6(5):883–891, 2012.

[40] S. B. Seidman and B. L. Foster. A graph theoretic generalization of the clique concept. *Journal of Mathematical Sociology*, 6:139–154, 1978.

[41] E. Tomita, Y. Sutani, T. Higashi, S. Takahashi, and M. Wakatsuki. A simple and faster branch-and-bound algorithm for finding a maximum clique. In M. Rahman and S. Fujita, editors, *WALCOM: Algorithms and Computation*, volume 5942 of *Lecture Notes in Computer Science*, pages 191–203. Springer Berlin Heidelberg, 2010.

[42] S. Trukhanov, C. Balasubramaniam, B. Balasundaram, and S. Butenko. Algorithms for detecting optimal hereditary structures in graphs, with application to clique relaxations. *Computational Optimization and Applications*, 56(1):113–130, 2013.

[43] A. Veremyev, O. Prokopyev, and E. Pasiliao. Critical nodes for communication efficiency and related problems in graphs. *Working Paper.*, 2014.

[44] B. Verweij, S. Ahmed, A. Kleywegt, G. Nemhauser, and A. Shapiro. The sample average approximation method applied to stochastic routing problems: A computational study. *Computational Optimization and Applications*, 24(2-3):289–333, 2003.

[45] A. Vinel and P. Krokhmal. Polyhedral approximations in p-order cone programming. *Optimization Methods and Software*, 29(6):1210–1237, 2014.

[46] A. G. Woodside and M. W. DeLozier. Effects of word of mouth advertising on consumer risk taking. *Journal of Advertising*, 5(4):12–19, 1976.

[47] M. Yannakakis. Node-and edge-deletion np-complete problems. In *STOC'78: Proceedings of the 10th Annual ACM Symposium on Theory of Computing*, pages 253–264, New York, 1978. ACM Press.