REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
29-02-2012	Final Report	03/2009- 11/2011

4. TITLE AND SUBTITLE
Probabilistic Analysis of Combinatorial Optimization Problems on Hypergraph Matchings

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-09-1-0088

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Krokhmal, Pavlo

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Iowa, 105 Jessup Hall, Iowa City, IA 52242-1316

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research
875 N Randolf Street
Arlington, VA 22023

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-OSR-VA-TR-2012-0947

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release

Distribution A

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This research project was concerned with probabilistic analysis of a class of discrete optimization problems whose underlying combinatorial structure is based on hypergraph matchings. The primary goal of was to provide theoretical insights into characteristic behavior and properties of this class of problems, which would furnish guidelines for development and tuning of solution algorithms, both exact and heuristic, as well as determine amenability of this class of problems to particular types of solution methods. We have established convergence limits for the optimal values of randomized hypergraph matching problems, along with the corresponding convergence rates, and proposed algorithms for finding solutions with guaranteed quality. For hypergraph matching problems with linear objectives, this task reduces to finding k-cliques in specially constructed k-partite graphs, for which a new branch-and-bound algorithm was developed that employs bit parallelism to achieve significant computational improvements over existing methods. In a related development, the behavior of quasi-cliques in random graphs was investigated, and it was ascertained that the size of the maximum quasi-clique undergoes a first-order phase transition, one of the first results of this kind.

15. SUBJECT TERMS
Probabilistic analysis, combinatorial optimization problems, random optimization problems, hypergraph matching problems, convergence rate, maximum clique problem, random graphs, clique relaxation, phase transition

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(Include area code)*

Probabilistic Analysis of Combinatorial Optimization Problems on Hypergraph Matchings

Final project report on AFOSR Grant FA9550-09-1-0088

PI: Pavlo A. Krokhmal
Department of Mechanical and Industrial Engineering
University of Iowa
3131 Seamans Center, Iowa City, IA 52242
E-mail: krokhmal@engineering.uiowa.edu

February 2012

1 Introduction

The assignment, or matching, problems, such as the Linear Assignment Problem (LAP) and the Quadratic Assignment Problem (QAP), are of theoretical and practical importance in a number of fields, from discrete mathematics and computer science to archeology and sports (for details on theory and applications of assignment problems see, among others, Pardalos and Pitsoulis, 2000; Pentico, 2007; Burkard et al., 2009). The underlying combinatorial structure of the feasible set common to these problems is that of a *perfect matching on a bi-partite graph*, i.e., a bijective mapping between the elements of two sets. The present endeavor is concerned with generalizations of the classical assignment problems where the underlying combinatorial structure is based on *hypergraph matchings*.

A hypergraph is a pair $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of vertices, and $\mathcal{E}$ is the set of hyperedges; a hyperedge $e \in \mathcal{E}$ is a subset of $\mathcal{V}$ containing two or more vertices (Berge, 1989; Bollobás, 1998). Similarly to regular graphs, a hypergraph $\mathcal{H}$ is called k-partite if its vertices can be partitioned into k independent subsets such that no vertices within a subset are connected by an edge. If all edges of a hypergraph contain exactly r vertices, the hypergraph is called r-uniform.

In this paper we restrict our discussion to d-partite d-uniform hypergraphs $\mathcal{H}_{d|n} = (\mathcal{V}, \mathcal{E})$, whose sets of vertices $\mathcal{V}$ are partitioned into d independent sets $\mathcal{V}_j$, each of cardinality n: $\mathcal{V} = \bigcup_{j=1}^{d} \mathcal{V}_j$, $|\mathcal{V}_j| = n$. Consequently, each hyperedge of $\mathcal{H}_{d|n}$ contains exactly one vertex from each independent set $\mathcal{V}_j$, and thus it can be represented as a vector $(i_1, \ldots, i_d) \in \mathcal{V}_1 \times \cdots \times \mathcal{V}_d$. This convention allows us to denote the vertices of each $\mathcal{V}_k$ as $1, 2, \ldots, n$, so that a hyperedge can be presented as $(i_1, \ldots, i_d) \in \{1, \ldots, n\}^d$. Finally, it is assumed that the d-partite d-uniform hypergraph $\mathcal{H}_{d|n}$ is *complete*, i.e., it contains n^d hyperedges.

Then, a (perfect) matching μ on $\mathcal{H}_{d|n}$ is formed by a set of n hyperedges that do not share any vertices, or, equivalently, which have the property that each vertex of the hypergraph belongs to exactly one hyperedge:

$$\mu = \big\{ \{e_1, \ldots, e_n\} \;\big|\; e_i \in \mathcal{E},\;\; e_i \cap e_j = \emptyset,\;\; 1 \le i, j \le n \big\}.$$

Since a hyperedge of $\mathcal{H}_{d|n}$ contains exactly one vertex from each of d independent sets $\mathcal{V}_k$, it can be represented as a vector $(i_1, \ldots, i_d) \in [\![n]\!]^d$, where the set $[\![n]\!] = \{1, \ldots, n\}$ is used to label the vertices of each $\mathcal{V}_k$. Then, the set $\mathcal{M}(\mathcal{H}_{d|n})$ of perfect matchings on $\mathcal{H}_{d|n}$ can be represented in mathematical programming form as

$$\mathcal{M}(\mathcal{H}_{d|n}) = \left\{ x \in \{0,1\}^{n^d} \;\middle|\; \sum_{\substack{i_k \in [\![n]\!] \\ k \in [\![d]\!] \setminus \{r\}}} x_{i_1 \cdots i_d} = 1, \;\; i_r \in [\![n]\!], \; r \in [\![d]\!] \right\}, \tag{1}$$

where $x_{i_1 \cdots i_d} = 1$ if the hyperedge $(i_1, \ldots, i_d)$ is included in the matching, and $x_{i_1 \cdots i_d} = 0$ otherwise.

If the cost of hypergraph matching μ is given by $\Phi(\mu)$, the general combinatorial optimization problem on hypergraph matchings can be formulated as

$$\min \left\{ \Phi(\mu) \;\middle|\; \mu \in \mathcal{M}(\mathcal{H}_{d|n}) \right\}, \tag{2}$$

where $\mathcal{M}(\mathcal{H}_{d|n})$ is the set of all perfect matchings on the hypergraph $\mathcal{H}_{d|n}$. A mathematical programming formulation of the general problem (2) can be written as

$$\begin{aligned}
\min \quad & \Phi(x) && && && \text{(3a)} \\
\text{s. t.} \quad & \sum_{i_2=1}^{n} \cdots \sum_{i_d=1}^{n} x_{i_1 \cdots i_d} = 1, && && i_1 = 1, \ldots, n, && \text{(3b)} \\
& \sum_{i_1=1}^{n} \cdots \sum_{i_{k-1}=1}^{n} \sum_{i_{k+1}=1}^{n} \cdots \sum_{i_d=1}^{n} x_{i_1 \cdots i_d} = 1, && i_k = 1, \ldots, n, && k = 2, \ldots, d-1, && \text{(3c)} \\
& \sum_{i_1=1}^{n} \cdots \sum_{i_{d-1}=1}^{n} x_{i_1 \cdots i_d} = 1, && && i_d = 1, \ldots, n, && \text{(3d)} \\
& x \in \{0,1\}^{n^d}, && && && \text{(3e)}
\end{aligned}$$

where $x_{i_1 \cdots i_d} = 1$ if the hyperedge $(i_1, \ldots, i_d)$ is included in the matching, and $x_{i_1 \cdots i_d} = 0$ otherwise.

The mathematical programming formulation (3) of the hypergraph matching problem (2) makes it natural to call problem (3) a *multidimensional assignment problem (MAP)*, where d stands for the number of "dimensions", and n is the number of "elements per dimension" (recall that d is equal to the number of independent subsets of vertices $\mathcal{V}_k$ in the hypergraph $\mathcal{H}_{d|n}$, and n denotes the number of vertices in each $\mathcal{V}_k$). In the sequel, we will use the terms "combinatorial optimization problem on hypergraph matchings" (or "hypergraph matching problem" for short) and "multidimensional assignment problem" in reference to (2)–(3) interchangeably.

Depending on the particular form of $\Phi(\cdot)$, a number of combinatorial optimization problems on hypergraph matchings can be formulated. In this paper, we analyze several problems that arise as special cases of (2)–(3) when the matching cost function Φ has the form

$$\Phi(\mu) = \coprod_{e_{i_1}, \ldots, e_{i_m} \in \mu} \phi_{e_{i_1} \cdots e_{i_m}}, \tag{4}$$

where $\coprod$ is an operator defined over some set of cost elements $\{\phi\}$ indexed by the hyperedges of $\mathcal{H}_{d|n}$. For instance, if the cost function in (2) is defined as $\Phi(\mu) = \sum_{e \in \mu} \phi_e$, or, equivalently, as a linear form over

variables $x_{i_1 \cdots i_d}$ in (3), one obtains the so-called Linear Multidimensional, or Multi-index Assignment Problem (LMAP):[1]

$$\begin{aligned} Z_{d|n} = \quad & \min \quad \sum_{i_1=1}^{n} \cdots \sum_{i_d=1}^{n} \phi_{i_1 \cdots i_d} \, x_{i_1 \cdots i_d} \\ & \text{s. t.} \quad \text{(3b)–(3e).} \end{aligned} \tag{5}$$

Taking $\Phi(x)$ as a quadratic form over $x \in \{0,1\}^{n^d}$, or, in other words, $\Phi(\mu) = \sum_{e_i, e_j \in \mu} \phi_{e_i e_j}$, leads to the Quadratic Multidimensional Assignment Problem (QMAP):

$$\begin{aligned} Q_{d|n} = \quad & \min \quad \sum_{i_1=1}^{n} \cdots \sum_{i_d=1}^{n} \sum_{j_1=1}^{n} \cdots \sum_{j_d=1}^{n} \phi_{i_1 \cdots i_d \, j_1 \cdots j_d} \, x_{i_1 \cdots i_d} \, x_{j_1 \cdots j_d} \\ & \text{s. t.} \quad \text{(3b)–(3e).} \end{aligned} \tag{6}$$

Similarly, if one is interested in minimizing the largest element of the corresponding linear or quadratic forms, the Linear Bottleneck MAP

$$\begin{aligned} W_{d|n} = \quad & \min \quad \max_{i_1, \ldots, i_d \in \{1, \ldots, n\}} \phi_{i_1 \cdots i_d} \, x_{i_1 \cdots i_d} \\ & \text{s. t.} \quad \text{(3b)–(3e)} \end{aligned} \tag{7}$$

and Quadratic Bottleneck MAP

$$\begin{aligned} U_{d|n} = \quad & \min \quad \max_{i_1, \ldots i_d, j_1, \ldots j_d \in \{1, \ldots, n\}} \phi_{i_1 \cdots i_d \, j_1 \cdots j_d} \, x_{i_1 \cdots i_d} \, x_{j_1 \cdots j_d} \\ & \text{s. t.} \quad \text{(3b)–(3e)} \end{aligned} \tag{8}$$

problems are obtained.

A matching $\mu = \{(i_1^{(1)}, \ldots, i_d^{(1)}), \ldots, (i_1^{(n)}, \ldots, i_d^{(n)})\}$ on $\mathcal{H}_{d|n}$ can be conveniently presented in the matrix form,

$$\mu = \begin{pmatrix} i_1^{(1)} & i_2^{(1)} & \cdots & i_d^{(1)} \\ i_1^{(2)} & i_2^{(2)} & \cdots & i_d^{(2)} \\ \vdots & \vdots & & \vdots \\ i_1^{(n)} & i_2^{(n)} & \cdots & i_d^{(n)} \end{pmatrix}, \tag{9}$$

where each column $\left(i_k^{(1)}, i_k^{(2)}, \ldots, i_k^{(n)}\right)^\top$, $k = 1, \ldots, d$, is a permutation of the set $[\![n]\!]$. Since the rows of matrix (9) represent hyperedges of $\mathcal{H}_{d|n}$, they can be permuted so that in the k-th dimension we have $\left(i_k^{(1)}, i_k^{(2)}, \ldots, i_k^{(n)}\right) = (1, \ldots, n)$. This allows for a permutation representation of a feasible solution of (2):

$$\mu = \begin{pmatrix} 1 & \pi_1(1) & \cdots & \pi_{d-1}(1) \\ 2 & \pi_1(2) & \cdots & \pi_{d-1}(2) \\ \vdots & \vdots & & \vdots \\ n & \pi_1(n) & \cdots & \pi_{d-1}(n) \end{pmatrix} = (\iota, \pi_1, \ldots, \pi_{d-1}), \tag{10}$$

[1] Problem (5) is also known in the literature as the *axial* MAP and is distinguished from the so-called *planar* MAP, whose underlying combinatorial structure is based on latin squares; see, e.g., Burkard et al. (2009).

where each $\pi_k : [\![n]\!] \mapsto [\![n]\!]$ is a permutation of the set $[\![n]\!]$, and ι is the identical permutation, $\iota(i) = i$, $i = 1, \ldots, n$.

The multidimensional assignment problems (5)–(8) also admit useful permutation formulations; for instance, the Linear MAP (5) can equivalently be written as

$$Z_{d|n} = \min_{\pi_1,\ldots,\pi_{d-1}\in\Pi_n} \sum_{i=1}^{n} \phi_{i\pi_1(i)\cdots\pi_{d-1}(i)}, \tag{11}$$

where π_k is a permutation of the set $[\![n]\!]$, and Π_n is the set of all such permutations. Problems (6)–(8) can be reformulated analogously. The permutation formulation makes it easy to see that the cardinality of the feasible set of (2) or, equivalently, (3), is

$$\left|\mathcal{M}(\mathcal{H}_{d|n})\right| = n!^{d-1}.$$

The Linear MAP (5) was first introduced by Pierskalla (1968), and had found numerous applications in the areas of data association, sensor fusion, multi-sensor multi-target tracking (Murphey et al., 1998a; Poore, 1994a; Kirubarajan et al., 2001; Poore and Gadaleta, 2006; Pusztaszeri et al., 1996; Andrijich and Caccetta, 2001; Chummun et al., 2001), image recognition (Veenman et al., 1998), and, recently, peer-to-peer refueling of space satellites (Dutta and Tsiotras, 2008); for a detailed discussion of the applications of the MAP, see, e.g., Burkard and Çela (1999a) and Burkard (2002). A three-dimensional version ($d = 3$) of the Quadratic MAP (6) was considered by Samra et al. (2005) and Hahn et al. (2008) in application to design of robust wireless transmission systems.

The objective of the present paper was to elucidate the properties of large-scale combinatorial optimization problems on hypergraph matchings (5)–(8) by analyzing the random instances of the corresponding problems, i.e., by assuming that the assignment costs $\phi_{i_1\cdots i_d\cdots}$ in (5)–(8) are iid random variables from a specified probability distribution. The introduced multidimensional assignment problems (5)–(8) constitute the primary focus of the present paper; as it will be seen below, the proposed techniques admit extension to other forms of cost function Φ given by (35) in the general formulation (2)–(3).

Related work Studies of random instances of combinatorial problems based on bipartite graph matchings that are given as special cases of (2)–(3) with $d = 2$ have been an active area of research for the last several decades. A number of important results have been obtained in the scope of random linear and quadratic assignment problems, some of them within the last few years; for a detailed review of random assignment problems see Krokhmal and Pardalos (2009).

Considerable attention has been paid to studies of the limiting behavior of the expected optimal value of the Linear Assignment Problem, a special case of (5) with $d = 2$. In the works of Walkup (1979); Karp (1987); Lazarus (1993); Goemans and Kodialam (1993); Olin (1992); Coppersmith and Sorkin (1999), a series of successively tighter upper and lower bounds for the expected cost $\mathsf{E}[Z_{2|n}]$ of LAP have been obtained in the assumption that the assignment costs are iid uniform on [0, 1] or exponential with mean 1. Bounds on $\mathsf{E}[Z_{2|n}]$ for general cost distributions have been furnished in Frenk et al. (1987), and Olin (1992) presents bounds for $Z_{2|n}$ that hold with probability 1.

Perhaps, the most widely known results concerning the random LAP are the conjectures by Mézard and Parisi (1985) and Parisi (1998) that the expected optimal cost of a random LAP with iid uniform [0, 1] or

exponential with mean 1 assignment costs satisfies

$$\lim_{n\to\infty} \mathsf{E}[Z_{2|n}] = \frac{\pi^2}{6} \approx 1.645 \qquad \text{and, moreover,} \qquad \mathsf{E}[Z_{2|n}] = \sum_{k=1}^{n} \frac{1}{k^2}. \tag{12}$$

The existence of the limit in (12) was argued by Mézard and Parisi (1985) using a non-rigorous replica method and was experimentally observed in Donath (1969) and Pardalos and Ramakrishnan (1993), but a theoretical proof of the limit in (12) was first furnished by Aldous (1992, 2001). Within the last five years, the second conjecture in (12) for finite-sized LAPs due to Parisi (1998) has been proven independently by Linusson and Wästlund (2004) and Nair et al. (2005), and its generalizations (Coppersmith and Sorkin, 1999; Linusson and Wästlund, 2000; Buck et al., 2002) for the case of random LAPs with non-square matrices (the so-called k-assignment problem) have been proven in Wästlund (2005a,b).

The methods of probabilistic analysis have played a pivotal role in explicating the computational properties of many hard combinatorial optimization problems, and, particularly, the Quadratic Assignment Problem that is given by (6) with $d = 2$. While the QAP is known to be NP-complete and non-approximable (Sahni and Gonzales, 1976) and thus very difficult to solve exactly (currently, instances of the QAP of sizes up to $n = 30$ can be solved rotinely, see Anstreicher, 2003; Loiola et al., 2007), it was noticed that good quality sub-optimal solutions of the QAP can be obtained relatively easily with many heuristic algorithms. It turns out that this strange behavior can be explained within the probabilistic framework, i.e., by analyzing the random instances of the QAP. In particular, Burkard and Fincke (1982a) were the first to point out that in a random QAP the ratio between the minimum (optimal) cost $Q_{2|n}$ and the maximum cost $Q^{2|n}$ approaches 1 in probability as the size of the problem increases:

$$\lim_{n\to\infty} \mathsf{P}\left\{\frac{Q^{2|n}}{Q_{2|n}} \le 1 + O\left(n^{-0.225}\right)\right\} = 1. \tag{13}$$

Frenk et al. (1985) and Rhee (1988, 1991) established stronger versions of (13), including a proof that the cost ratio in (13) converges to unity not only in probability, but almost surely (a.s.). It must be noted that the remarkable observation of Burkard and Fincke (1982a) has led to the discovery of an entire class of combinatorial optimization problems with similar behavior (Burkard and Fincke, 1985; Szpankowski, 1995; Albrecher et al., 2006).

A number of results along the lines of (12) and (13) have been obtained for other combinatorial optimization problems based on bipartite graph matchings, such as linear and quadratic bottleneck assignment problems, bi-quadratic assignment problem, etc., in Pferschy (1996); Burkard et al. (1994); Burkard and Fincke (1982b); Albrecher (2005), and others; see Krokhmal and Pardalos (2009) for details.

As regards the combinatorial optimization problems on hypergraph matchings (2)–(3), relatively few attempts have been made in the literature to investigate their properties in large-scale instances. Dyer, Frieze, and McDiarmid (1986) derived an upper bound on the expected optimal cost $\mathsf{E}[Z^{\mathrm{LP}}_{d|n}]$ of the linear programming relaxation of random LMAP (5) under the assumption that hyperedge costs $\phi_{i_1\cdots i_d}$ are iid uniform on $[0, 1]$:

$$\mathsf{E}[Z^{\mathrm{LP}}_{d|n}] \le d/n^{d-2}.$$

More recently, random instances of the Linear MAP (5) have been analyzed using the methods of statistical physics by Martin, Mézard, and Rivoire (2005) in an attempt to generalize the corresponding conjecture (12) of Mézard and Parisi (1985) for the random LAP; unfortunately, the approach employed

by the authors did not yield an explicit expression for the expected optimal cost of random Linear MAP (5).

Krokhmal et al. (2007) have established a limiting value of the expected optimal cost $\mathsf{E}[Z_{d|n}]$ of random Linear MAP (5) with absolutely continuous cost distributions F that admit a power series asymptotical expansion in the neighborhood of the left-end point $F^{-1}(0)$ of their support sets:

$$\lim_{n \text{ or } d\to\infty} \frac{1}{n}\mathsf{E}[Z_{d|n}] = F^{-1}(0) \equiv \inf\{\, t \mid F(t) > 0 \,\}. \tag{14}$$

2 On optimality of a polynomial algorithm for random linear multidimensional assignment problem

The Linear Multidimensional Assignment Problem (LMAP) is a higher-dimensional generalization of the well-known two-dimensional, or Linear Assignment Problem (LAP) (see, e.g., Papadimitrou and Steiglitz, 1998; Burkard et al., 2009). A graph-theoretic formulation of the LAP of cardinality n presents it as finding a minimum-cost perfect matching on a balanced bipartite graph with $2n$ vertices, provided that the cost of matching is defined as the sum of edge costs. Similarly, a d-dimensional LMAP of cardinality n can be formulated as finding a perfect matching on a balanced d-partite d-uniform hypergraph with dn vertices, such that the sum of the costs of hyperedges in the matching is minimized (see, among others, Berge, 1989; Bollobás, 1998 for general references on hypergraphs). If the cost of hyperedge $(i_1, \ldots, i_d)$, where $i_1, \ldots, i_d \in \{1, \ldots, n\}$, is given by $\phi_{i_1 \cdots i_d}$, the mathematical programming formulation of the LMAP of dimensionality d and cardinality n reads as

$$\begin{aligned}
\min \quad & \sum_{i_1=1}^{n} \cdots \sum_{i_d=1}^{n} \phi_{i_1 \cdots i_d}\, x_{i_1 \cdots i_d} \\
\text{s. t.} \quad & \sum_{i_2=1}^{n} \cdots \sum_{i_d=1}^{n} x_{i_1 \cdots i_d} = 1, && i_1 = 1, \ldots, n, \\
& \sum_{i_1=1}^{n} \cdots \sum_{i_{k-1}=1}^{n} \sum_{i_{k+1}=1}^{n} \cdots \sum_{i_d=1}^{n} x_{i_1 \cdots i_d} = 1, && \begin{array}{l} i_k = 1, \ldots, n, \\ k = 2, \ldots, d-1, \end{array} \\
& \sum_{i_1=1}^{n} \cdots \sum_{i_{d-1}=1}^{n} x_{i_1 \cdots i_d} = 1, && i_d = 1, \ldots, n, \\
& x_{i_1 \cdots i_d} \in \{0, 1\}, && i_k = 1, \ldots, n,\ k = 1, \ldots, d,
\end{aligned} \tag{15}$$

where $x_{i_1 \cdots i_d} = 1$ if the hyperedge $(i_1, \ldots, i_d)$ is included in the matching, and $x_{i_1 \cdots i_d} = 0$ otherwise. It is easy to see that a feasible solution of (15) is given by n hyperedges $\left(i_1^{(r)}, \ldots, i_d^{(r)}\right)$, $r = 1, \ldots, n$, such that in each dimension k the set $\left\{i_k^{(1)}, \ldots, i_k^{(n)}\right\}$ is a permutation of $\{1, \ldots, n\}$. Hence, problem (15) admits the following geometric interpretation: given a d-dimensional matrix $\Phi = \{\phi_{i_1 \cdots i_d}\} \in \mathbb{R}^{n^d}$, find such a permutation of its "rows" and "columns" in all dimensions that the sum of the diagonal elements is minimized, and thus is also known in the literature as "axial" multidimensional, or multi-index assignment problem.

The LMAP (15) was first introduced by Pierskalla (1968), and has found numerous applications in the areas of data association, image recognition, multi-sensor multi-target tracking, peer-to-peer satellite

refueling, and so on (for a detailed discussion of the properties and applications of the LMAP, see, for example, Burkard and Çela, 1999b; Burkard, 2002; Burkard et al., 2009).

In contrast to the LAP that is polynomially solvable, the LMAP with $d \geq 3$ is generally NP-complete, a fact first established by Karp (1972) for the 3-dimensional ($d = 3$) assignment problem (15). For a discussion of approximation properties of the LMAP (15), see, e.g., Spieksma (2000); in particular, Crama and Spieksma (1992) have shown that even in the case when costs $\phi_{i_1 i_2 i_3}$ of the 3-dimensional LMAP are decomposable, i.e., $\phi_{i_1 i_2 i_3} = d_{i_1 i_2} + d_{i_2 i_3} + d_{i_1 i_3}$, there is no polynomial algorithm that yields an ε-approximate solution for any $\varepsilon > 0$.

Exact and heuristic algorithms for three- and higher-dimensional LMAPs were proposed in Balas and Saltzman (1991); Poore (1994a,b); Poore and Robertson (1997); Murphey et al. (1998a,b); Andrijich and Caccetta (2001), and others. In particular, Balas and Saltzman (1991) introduced a number of heuristics for the 3-dimensional LMAP; Andrijich and Caccetta (2001) report that on randomly generated problems, some of these heuristics yield solutions very close to optimality. These observations find a theoretical substantiation in Kravtsov (2005), who has demonstrated that if the assignment costs in (15) are iid random variables from a discrete distribution satisfying certain properties, a simple greedy algorithm produces asymptotically optimal solutions with high probability, when the cardinality n of the LMAP (15) increases infinitely.

In this work, we strengthen and generalize the results of Kravtsov (2005), by showing that a greedy algorithm produces ε-approximate solutions of random LMAP *almost surely* (a. s.), or, in other words, that the cost of the greedy solution converges strongly to the optimal cost. Further, we extend the analysis to random LMAPs whose costs are continuously distributed, including distributions with unbounded support sets.

Results concerning asymptotic optimality of heuristic algorithms on randomly generated problems are well known in the context of other hard combinatorial optimization problems, such as the Quadratic Assignment Problem (QAP), which is also known to be NP-complete and non-approximable (see, among others, Burkard et al., 2009; Krokhmal and Pardalos, 2009, and references therein). In the case of random QAP, asymptotic optimality of heuristic solution methods is a manifestation of the fact that for instances of random QAP large enough, *all* its feasible solutions are asymptotically optimal (Burkard and Fincke, 1982a). Moreover, an entire class of combinatorial optimization problems exists that shares this property with the QAP (Burkard and Fincke, 1985; Szpankowski, 1995). The LMAP (15), however, does *not* belong to this class; recent investigations of asymptotic behavior of random LMAPs (Krokhmal et al., 2007; Krokhmal and Pardalos, 2011) entail that only a vanishingly small fraction the feasible set of a random LMAP is ε-optimal.

2.1 A greedy algorithm for LMAP with discrete iid random costs

Algorithm 1 describes the greedy heuristic for the LMAP that is in the focus of this work. The heuristic starts by finding the smallest hyperedge cost of the LMAP (15), which we denote by $\phi_{i_1^{(1)} \dots i_d^{(1)}}$, and then removing from the cost matrix Φ the cost elements $\phi_{i_1 \dots i_\ell^{(1)} \dots i_d}$ for each $\ell \in \{1, \dots, d\}$, i.e., the costs of the hyperedges that are not feasible with respect to the smallest-cost hyperedge $\left(i_1^{(1)}, \dots, i_d^{(1)}\right)$. Then, the procedure is repeated, and upon finding the smallest hyperedge cost $\phi_{i_1^{(2)} \dots i_d^{(2)}}$ in the reduced cost array Φ, the costs of hyperedges that are infeasible with respect to the hyperedges $\left(i_1^{(k)}, \dots, i_d^{(k)}\right)$, $k = 1, 2$, are discarded, and so on. After n steps, n costs $\phi_{i_1^{(k)} \dots i_d^{(k)}}$ are obtained, which have the property that for

each $\ell = 1, \ldots, d$ the indices $\{i_\ell^{(1)}, \ldots, i_\ell^{(n)}\}$ are all different, i.e., a feasible solution of (15) is found.

Algorithm 1 A greedy heuristic for LMAP (15)

1: **input:** Cost matrix $\Phi = \{\phi_{i_1 \cdots i_d} \mid (i_1, \ldots, i_d) \in \{1, \ldots, n\}^d\} \in \mathbb{R}^{n^d}$

2: **initialize:** $\widetilde{Z}_n := 0$; for each $\ell \in \{1, \ldots, d\}$ define set $\mathcal{N}_\ell := \{1, \ldots, n\}$

3: **for** $k := 1$ **to** n **do**

4: define a submatrix $\Phi^{(k)} \in \mathbb{R}^{(n-k+1)^d}$ of the cost matrix Φ as $\Phi^{(k)} := \{\phi_{i_1 \cdots i_d} \mid (i_1, \ldots, i_d) \in \mathcal{N}_1 \times \cdots \times \mathcal{N}_d\}$

5: find the smallest element $\phi_{i_1^{(k)} \cdots i_d^{(k)}}$ of the submatrix $\Phi^{(k)}$: $\left(i_1^{(k)}, \ldots, i_d^{(k)}\right) \in \arg\min \{\phi_{i_1 \cdots i_d} \in \Phi^{(k)}\}$

6: let $\widetilde{Z}_n := \widetilde{Z}_n + \phi_{i_1^{(k)} \cdots i_d^{(k)}}$

7: for each $\ell \in \{1, \ldots, d\}$ update the set $\mathcal{N}_\ell := \mathcal{N}_\ell \backslash \{i_\ell^{(k)}\}$

8: **end for**

9: for each $k \in \{1, \ldots, n\}$ define $\tilde{x}_{i_1^{(k)} \cdots i_d^{(k)}} := 1$ and $\tilde{x}_{i_1 \cdots i_d} := 0$ for all other $(i_1, \ldots, i_d)$

10: **output:** A feasible solution $\tilde{x}_{i_1 \cdots i_d}$ of LMAP (15) and its cost $\widetilde{Z}_n$

Obviously, the described greedy heuristic for LMAP runs in $O(n^{d+1})$ time. The next lemma provides a foundation for the subsequent probabilistic analysis of the greedy heuristic and is a strengthening of the corresponding result in Kravtsov (2005).

Lemma 1. *Consider a set $\mathcal{S}_n$ of cardinality $|\mathcal{S}_n| = \kappa_n$ whose elements are iid random variables distributed uniformly over ρ_n values $a_n < \ldots < b_n$. Assume that ρ_n and κ_n increase with n such that the following series converges:*

$$\sum_n \rho_n e^{-\frac{\kappa_n}{\rho_n}} < \infty.$$

Then, for n sufficiently large, the set $\mathcal{S}_n$ contains the minimum element a_n almost surely (a.s.)

Proof. To verify the statement of the lemma, it is convenient to think about the set $\mathcal{S}_n$ in terms of randomly distributing $|\mathcal{S}_n| = \kappa_n$ different objects into ρ_n boxes. Then, define $\mathcal{A}_n$ as the event that $\mathcal{S}_n$ contains the smallest element a_n, whence

$$\mathsf{P}\{\bar{\mathcal{A}}_n\} \le \mathsf{P}\{\text{at least one box is empty}\} = 1 - \mathsf{P}\{\mathcal{B}_n\}, \tag{16}$$

where $\mathcal{B}_n$ is the event that there are no empty boxes, for which it holds (see, e.g., Feller, 1968):

$$\mathsf{P}\{\mathcal{B}_n\} = \sum_{i=0}^{\rho_n} (-1)^i \binom{\rho_n}{i} \left(1 - \frac{i}{\rho_n}\right)^{\kappa_n}.$$

If ρ_n and κ_n increase with n such that the quantity $\lambda_n = \rho_n e^{-\frac{\kappa_n}{\rho_n}}$ is bounded, that it can be shown that each summand in the above sum is asymptotically equal to $(-\lambda_n)^i / i!$, whereby

$$\mathsf{P}\{\mathcal{B}_n\} \to e^{-\lambda_n}, \quad n \to \infty.$$

Thus, for n sufficiently large, the probability that the set $\mathcal{S}_n$ does not contain the smallest element a_n of the distribution can be bounded as

$$\mathsf{P}\{\bar{\mathcal{A}}_n\} \le 1 - \mathsf{P}\{\mathcal{B}_n\} \approx 1 - e^{-\lambda_n} = \lambda_n + O(\lambda_n^2).$$

Since by the conditions of the Lemma, $\sum_n \lambda_n < \infty$, from the Borel-Cantelli lemma we immediately have that $\mathsf{P}\{\bar{\mathcal{A}}_n \text{ i.o.}\} = 0 \;\Leftrightarrow\; \mathsf{P}\{\mathcal{A}_n \text{ ev.}\} = 1$. ■

Assuming that the assignment costs of the LMAP (15) are positive, a feasible solution with cost $\widetilde{Z}_n$ is an ε-approximation of the optimal cost Z_n^* of LMAP of cardinality n if it satisfies

$$\widetilde{Z}_n \le Z_n^*(1+\varepsilon). \tag{17}$$

The next theorem establishes the conditions on the discrete distribution of assignment costs in (15) under which the greedy algorithm delivers an ε-approximation of the optimal cost of the LMAP, or, more precisely, the ratio of the greedy solution cost to the optimal cost approaches unity almost surely.

Theorem 1. *Consider LMAP* (15) *with $d \ge 3$, $n \ge 2$, whose cost coefficients are iid random variables distributed uniformly over n^α values*[2] *$a_n < \ldots < b_n$, where $a_n > 0$ and $\alpha > 0$. Then, there exists a constant $M > 0$ such that the greedy algorithm produces a solution with the cost $\widetilde{Z}_n$, which for sufficiently large n satisfies*

$$\frac{\widetilde{Z}_n}{Z_n^*} - 1 \le M\left(\frac{b_n}{a_n} - 1\right) n^{\alpha/d-1} \ln^{1/d} n \qquad a.\,s., \tag{18}$$

where Z_n^* is the optimal cost of the LMAP.

Proof. From the description of the greedy heuristic, it follows that the cost of the feasible solution can be represented as

$$\widetilde{Z}_n = \sum_{k=1}^{n} \phi_{i_1^{(k)}\ldots i_d^{(k)}} = \sum_{k=1}^{n} \tilde{\phi}_k, \tag{19}$$

where each $\tilde{\phi}_k$ is equal to the smallest element of a submatrix $\Phi^{(k)}$:

$$\tilde{\phi}_k = \min\{a \mid a \in \Phi^{(k)}\}, \quad k = 1, \ldots, n.$$

In general, the summation in (19) contains terms $\tilde{\phi}_k$ that are either equal to the smallest element a_n of the distribution, or exceed it. Let K_n denote the (random) number of the summands in (19) that are greater than a_n:

$$K_n = \big|\{k \mid \tilde{\phi}_k > a_n\}\big|. \tag{20}$$

Then, noting that $0 \le K_n \le n$, the optimal cost Z_n^* of the LMAP (15) and the cost $\widetilde{Z}_n$ returned by the greedy heuristic can be bounded as

$$n a_n \le Z_n^* \le \widetilde{Z}_n \le (n - K_n)a_n + K_n b_n = n a_n \left(1 + \frac{K_n}{n}\frac{b_n - a_n}{a_n}\right), \tag{21}$$

[2]Here and in what follows we omit rounding to avoid unnecessary ramifications in exposition.

from which it is easy to see that $\widetilde{Z}_n$ is an ε-approximate solution of (15) by means of the approximation inequality (17) as soon as $\widetilde{Z}_n \leq n a_n(1+\varepsilon)$. Thus, for some $\varepsilon_n > 0$ consider

$$\mathsf{P}\{\widetilde{Z}_n > n a_n(1+\varepsilon_n)\} \leq \mathsf{P}\{K_n > \gamma_n n\},$$

where the inequality follows from (21) provided that γ_n is chosen as

$$\gamma_n = \frac{\varepsilon_n}{b_n/a_n - 1}.$$

Observe that if $K_n > \gamma_n n$ holds, then there exists an integer $\nu \in \{1, \ldots, n\}$ such that $\nu > \gamma_n n$ and the corresponding submatrix of Φ with ν^d elements does not contain elements equal to a_n. Then, from Lemma 4 it follows that for sufficiently large values of n,

$$\mathsf{P}\{K_n > \gamma_n n\} \leq \mathsf{P}\{\text{ set of size } (\gamma_n n)^d \text{ does not contain } a_n \} \leq n^{\alpha} \exp\left\{-\frac{(\gamma_n n)^d}{n^{\alpha}}\right\}.$$

Choosing the parameter ε_n in the form

$$\varepsilon_n = (2+\alpha)^{1/d} \left(\frac{b_n}{a_n} - 1\right) n^{\alpha/d - 1} \ln^{1/d} n,$$

we have that for values of n large enough,

$$\mathsf{P}\{K_n > \gamma_n n\} \leq n^{-2},$$

whence expression (18) follows by the Borel-Cantelli lemma. ■

Corollary 1.1. *If $\alpha < d$ in* (18) *and the ratio b_n/a_n satisfies*

$$\frac{b_n}{a_n} = o\left(n^{1-\alpha/d} \ln^{-1/d} n\right), \quad n \gg 1,$$

then for sufficiently large n, the greedy cost $\widetilde{Z}_n$ is an ε-approximation of the optimal cost Z_n^* of random LMAP due to (17) *for any $\varepsilon > 0$ almost surely. Put differently, the cost ratio $\widetilde{Z}_n / Z_n^*$ between greedy and optimal solutions converges to unity a. s., with the convergence rate given by* (18)*.*

Remark 1.1. The intuition behind Lemma 4 and Theorem 5 is that if the elements of the cost matrix Φ are drawn at random from a sufficiently small set of values, then at each step of the greedy heuristic, the submatrix $\Phi^{(k)}$ will contain the smallest element from that set with sufficiently high probability. This observation can be pressed into service to address the case when the elements of the cost matrix Φ of the LMAP (15) are continuous iid variables, as it is shown next.

2.2 Greedy heuristic for LMAP with continuous iid costs

In (Krokhmal et al., 2007; Krokhmal and Pardalos, 2011) it has been demonstrated that if the assignment costs $\phi_{i_1 \cdots i_d}$ in LMAP (15) are iid random variables with a continuous distribution F, then asymptotic behavior of the optimal value of random LMAP is controlled by the properties of the distribution F in the vicinity of the left-end point $F^{-1}(0)$ of the support set of the distribution, where

$$F^{-1}(0) = \inf\{t \mid F(t) > 0\}.$$

In view of that, we restrict our discussion to continuous distributions F whose support sets are bounded from above,

$$F^{-1}(1) < +\infty, \quad \text{where} \quad F^{-1}(1) = \sup\{t \mid F(t) < 1\}.$$

The next theorem generalizes the results of the previous section to continuous distributions.

Theorem 2. *Consider LMAP* (15) *with $d \geq 3$, $n \geq 2$, whose cost coefficients are iid random variables with a continuous distribution F that has a bounded support $[a, b]$, where $a > 0$. Then, for any $\alpha > 0$, there exists a constant $M > 0$ such that the greedy algorithm produces a solution with cost $\widetilde{Z}_n$, which for sufficiently large n satisfies*

$$\frac{\widetilde{Z}_n}{Z_n^*} - 1 \leq \frac{F^{-1}(n^{-\alpha})}{a} - 1 + M n^{\alpha/d - 1} \ln^{1/d} n \qquad a.\,s., \tag{22}$$

where Z_n^* is the optimal cost of the LMAP.

Proof. For a continuous distribution F on $[a, b] \subset \mathbb{R}$, define the sequence $\{\delta_n(k)\}$, $k = 0, \ldots, \rho_n$, as

$$\delta_n(0) = 0, \quad \delta_n(k) = -a + F^{-1}\left(\frac{1}{\rho_n} + F(a + \delta_n(k-1))\right), \quad k = 1, \ldots, \rho_n.$$

The intervals $\mathcal{I}_k = (a + \delta_n(k-1), a + \delta_n(k)]$, $k = 1, \ldots, \rho_n$, partition the set $(a, b]$ into ρ_n equiprobable "bins", such that for any F-distributed random variable X

$$\mathsf{P}\{a + \delta_n(k-1) < X \leq a + \delta_n(k)\} = \frac{1}{\rho_n}, \quad k = 1, \ldots, \rho_n.$$

Then, the elements of the cost matrix Φ can be labeled with ρ_n different labels, in accordance to the "bin" $\mathcal{I}_k$ that the corresponding cost element falls into. Obviously, the labels are independently and identically uniformly distributed. Therefore, taking into account that the elements of the cost matrix Φ that fall into bin $\mathcal{I}_1$ are less than or equal to $a + \delta_n(1)$, the cost $\widetilde{Z}_n$ of the greedy solution of the MAP can be bounded as

$$n a \leq Z_n^* \leq \widetilde{Z}_n \leq (n - K_n)(a + \delta_n(1)) + K_n b, \tag{23}$$

where K_n is equal to the number of summands in the cost $\widetilde{Z}_n$ of the greedy solution that do not fall into the bin $\mathcal{I}_1 = (a, a + \delta_n(1)]$. Then, for any fixed $\varepsilon_n > 0$ it holds that

$$\mathsf{P}\{\widetilde{Z}_n(\Phi) > n a(1 + \varepsilon_n)\} \leq \mathsf{P}\{(n - K_n)(a + \delta_n(1)) + K_n b > n a(1 + \varepsilon_n)\} = \mathsf{P}\{K_n > n\gamma_n\},$$

where

$$\gamma_n = \frac{\varepsilon_n - \delta_n(1)/a}{b/a - 1 - \delta_n(1)/a}.$$

Similarly to the arguments of Theorem 5, $K_n > n\gamma_n$ holds provided that there exists $\nu \in \{1, \ldots, n\}$ such that $\nu > n\gamma_n$ and the corresponding submatrix of size ν^d does not contain elements from the interval $\mathcal{I}_1$, whence

$$\mathsf{P}\{K_n > \gamma_n n\} \leq \rho_n \exp\left\{-\frac{(\gamma_n n)^d}{\rho_n}\right\},$$

Let $\rho_n = n^\alpha$ for some $\alpha > 0$, then, choosing ε_n as

$$\varepsilon_n = \frac{F^{-1}(n^{-\alpha})}{a} - 1 + \left(\frac{b}{a} - \frac{F^{-1}(n^{-\alpha})}{a}\right)(\alpha+2)^{1/d} n^{\alpha/d-1} \ln^{1/d} n,$$

and taking into account that $\delta_n(1) = F^{-1}(n^{-\alpha}) - a = o(1)$, $n \gg 1$, we obtain that

$$\mathsf{P}\{K_n > \gamma_n n\} \le \rho_n^{-2/\alpha} = n^{-2},$$

which verifies statement (22) of the Theorem by virtue of the Borel-Cantelli lemma. ■

Corollary 2.1. *If $\alpha < d$, it follows from* (22) *that $\widetilde{Z}_n$ represents an ε-optimal solution of random LMAP in the sense* (17) *for any $\varepsilon > 0$, and the cost ratio $\widetilde{Z}_n / Z_n^*$ converges to unity a. s. It is natural that the value of the parameter $\alpha \in (0, d)$ in* (22) *is selected based on the properties of F^{-1} at the origin so as to increase the rate of convergence. In particular, if in some neighborhood of 0 the inverse F^{-1} of the distribution F satisfies for some $\nu > 0$*

$$F^{-1}(u) \le a + Lu^\nu, \quad L > 0,\ u \to 0+,$$

then there exists a constant $M_1 > 0$ such that

$$\frac{\widetilde{Z}_n}{Z_n^*} - 1 \le M_1 n^{(1+\nu d)^{-1}-1} \ln^{1/d} n \qquad a.\,s.$$

Next, we consider the case of a continuous distribution F with support of the form $(-\infty, b]$, where the following bounds on the optimal cost of random LMAP play a key role. Namely, as shown in Krokhmal and Pardalos (2011), the optimal value Z_n^* of random LMAP with iid cost coefficients whose distribution has a support unbounded from below, satisfies for sufficiently large n

$$n F^{-1}\left(\frac{1}{n^{d-1}}\right) \le Z_n^* \le n F^{-1}\left(\frac{3\ln n}{n^{d-1}}\right) \qquad \text{a. s.} \tag{24}$$

Expression (24) entails that when support of F is unbounded from below, $F^{-1}(0) = -\infty$, one has that $Z_n^* < 0$ a. s. for large enough n. Note that in this case the approximation condition (17) takes the form

$$\widetilde{Z}_n \le Z_n^*(1-\varepsilon), \quad \varepsilon > 0. \tag{25}$$

Taking into account (25), the following statement holds regarding the quality of the greedy solution to a random LMAP (15).

Theorem 3. *Consider LMAP* (15) *with $d \ge 3$, $n \ge 2$ whose cost coefficients are iid random variables with continuous distribution F such that $F^{-1}(0) = -\infty$, $F^{-1}(1) < \infty$. Then, the greedy algorithm produces a solution with cost $\widetilde{Z}_n$ that for sufficiently large n satisfies*

$$1 - \frac{\widetilde{Z}_n}{Z_n^*} \le \left(\frac{d-1}{n}\right)^{1/d} + \frac{F^{-1}\left(\frac{1}{n^{d-1}}\right)}{F^{-1}\left(\frac{3\ln n}{n^{d-1}}\right)} - 1 \qquad a.\,s., \tag{26}$$

where Z_n^* is the optimal cost of the LMAP.

Proof. Similarly to the proof of Theorem 2, let us partition the semi-infinite support $(-\infty, b]$ of the distribution F into ρ_n "bins" $(\alpha_n(k-1), \alpha_n(k)]$ such that

$$\mathsf{P}\{\alpha_n(k-1) < X \le \alpha_n(k)\} = \frac{1}{\rho_n}, \quad k = 1, \ldots, \rho_n,$$

where X is F-distributed random variable, and $\alpha_n(k)$ is defined as

$$\alpha_n(0) = -\infty, \quad \alpha_n(k) = F^{-1}\left(F(\alpha_n(k-1)) + \frac{1}{\rho_n}\right), \quad k = 1, \ldots, \rho_n. \tag{27}$$

Then, similar arguments allow us to construct an upper bound $\widetilde{\widetilde{Z}}_n$ on the greedy cost $\widetilde{Z}_n$ in the form

$$\widetilde{Z}_n \le \widetilde{\widetilde{Z}}_n \equiv (n - K_n)F^{-1}(\rho_n^{-1}) + K_n F^{-1}(1),$$

where N_k is the number of summands in the greedy cost $\widetilde{Z}_n$ that do not fall into the first "bin" $(-\infty, F^{-1}(\rho_n^{-1})]$.

Next, observe that if the optimal cost Z_n^* of the LMAP can be bounded from below and above, e.g.,

$$\underline{Z}_n \le Z_n^* \le \bar{Z}_n,$$

then, given a fixed $\varepsilon > 0$, the greedy solution cost $\widetilde{Z}_n$ satisfies the approximation inequality (25) as soon as the upper bound $\widetilde{\widetilde{Z}}_n$ satisfies

$$\widetilde{\widetilde{Z}}_n - \underline{Z}_n \le -\varepsilon \bar{Z}_n.$$

In view of the bounds (24) on the optimal cost of random LMAP due to (Krokhmal and Pardalos, 2011) that hold almost surely for large enough n, define

$$\underline{Z}_n = nF^{-1}\left(\frac{1}{3\rho_n \ln n}\right), \quad \bar{Z}_n = nF^{-1}\left(\frac{1}{\rho_n}\right), \quad \text{where} \quad \rho_n = \frac{n^{d-1}}{3\ln n},$$

and for any fixed $\varepsilon_n > 0$ consider the probability

$$\mathsf{P}\left\{\widetilde{\widetilde{Z}}_n(\Phi) - \underline{Z}_n > -\varepsilon_n \bar{Z}_n\right\} = \mathsf{P}\{K_n > n\gamma_n\}, \tag{28}$$

where

$$\gamma_n = \left(F^{-1}\left(\frac{1}{3\rho_n \ln n}\right) - F^{-1}\left(\frac{1}{\rho_n}\right)(1+\varepsilon_n)\right)\left(F^{-1}(1) - F^{-1}\left(\frac{1}{\rho_n}\right)\right)^{-1}.$$

Following the arguments of Theorems 5 and 2, it can be shown that for sufficiently large values of n the probability in (28) satisfies

$$\mathsf{P}\{K_n > n\gamma_n\} \le \rho_n \exp\left\{-\frac{(n\gamma_n)^d}{\rho_n}\right\}. \tag{29}$$

If one selects the parameter ε_n in the form

$$\varepsilon_n = \frac{F^{-1}\left(\dfrac{1}{3\rho_n \ln n}\right)}{F^{-1}\left(\rho_n^{-1}\right)} + \left(1 - \frac{F^{-1}(1)}{F^{-1}\left(\rho_n^{-1}\right)}\right) \frac{1}{n}\left(3\rho_n \ln \rho_n\right)^{1/d} - 1$$

$$= \left(\frac{d-1}{n}\right)^{1/d} \left(1 - \frac{\ln(3\ln n)}{(d-1)\ln n}\right)^{1/d} + \frac{F^{-1}\left(\dfrac{1}{3\rho_n \ln n}\right)}{F^{-1}\left(\rho_n^{-1}\right)} - 1,$$

inequality (29) implies that the probability in (28) is bounded as

$$\mathsf{P}\{K_n > n\gamma_n\} \le \rho_n^{-2} \le n^{-2}$$

for all large enough values of n, thereby verifying the estimate (26) of approximation quality of the greedy solution by means of the Borel-Cantelli lemma. ■

Remark 3.1. In the case when the distribution F is such that

$$F^{-1}\left(\frac{1}{n^{d-1}}\right) \Big/ F^{-1}\left(\frac{3\ln n}{n^{d-1}}\right) \to 1, \quad n \to \infty, \tag{30}$$

Theorem 3 asserts that the solution cost produced by the greedy heuristic is an ε-approximation of the optimal cost of random LMAP, for any $\varepsilon > 0$. Condition (30) holds, for instance, for distributions F whose inverse F^{-1} has a logarithmic singularity at the origin, i.e., when the following asymptotic representation holds in the vicinity of 0:

$$F^{-1}(u) \sim -c_0 \ln^{\beta} \frac{1}{u}, \quad u \to 0+ \quad \text{for some } \ c_0 > 0, \ \beta > 0.$$

Continuous distributions that satisfy this condition and whose support is bounded from above include, for instance, exponential distribution on $(-\infty, 0]$, truncated normal distribution on $(-\infty, b]$:

$$F(t) = e^t \mathbf{1}_{(-\infty,0]}(t) + \mathbf{1}_{(0,\infty)}(t), \quad F(t) = \frac{\Phi(t)}{\Phi(b)} \mathbf{1}_{(-\infty,b]}(t) + \mathbf{1}_{(b,\infty)}(t),$$

where $\Phi(t)$ is the standard normal distribution function. Observe that in the case when the inverse F^{-1} of the cost distribution F has, for example, a power singularity at the origin, i.e.,

$$F^{-1}(u) \sim -c_0 u^{-\beta}, \quad u \to 0+, \quad c_0, \beta > 0,$$

the ratio in (30) is unbounded in n, hence no statement can be inferred from Theorem 3 regarding the quality of the greedy solution in this case.

Remark 3.2. According to Krokhmal and Pardalos (2011), the asymptotic behavior of the optimal cost of random LMAP (15) is determined completely by the properties of the distribution function F in the vicinity of the left-end point of its support. Nevertheless, the requirement of boundedness from above of the distribution's support, $F^{-1}(1) < \infty$, which is imposed in Theorems 5–3, is essential for estimating the quality of the greedy solution, as it allows one to obtain an upper bound on the cost of the solution produced by the greedy algorithm.

3 On Hamming distance in hypergraph matching problems

The concept of *distance* between feasible solutions of optimization problems plays an important role in combinatorial optimization. For instance, it is widely acknowledged that analysis of the problem's *fitness landscape* (Weinberger, 1990; Stadler, 1996), which comprises the set of feasible solutions, their fitness values (costs) $f(\cdot)$, and a measure $d(\cdot\,,\cdot)$ of distance between solutions, can yield useful insights into the performance and tuning of exact and heuristic algorithms.

This section discusses some properties of the Hamming distance, a popular distance measure $d(\cdot\,,\cdot)$ in combinatorial optimization, in application to problems where the underlying combinatorial structure is based on *hypergraph matchings*, which generalize the well-known class of combinatorial problems on bipartite graph matchings, such as the linear assignment problem (LAP), quadratic assignment problem (QAP), etc. (comprehensive reviews of assignment problems can be found in, e.g., Pardalos and Pitsoulis, 2000; Burkard, 2002; Burkard et al., 2009).

The Hamming distance was first introduced by Hamming (1950) as a measure of errors (or substitutions) that transform one string of a binary code into another, and since then has found applications in the areas of coding theory, information theory, cryptography, combinatorial optimization, and others (Matsumoto et al., 2006). Given two strings of equal length with characters from any alphabet (not necessarily binary), the Hamming distance between them is usually defined as the number of positions in which these strings disagree. The landscape structure of many combinatorial optimization problems can be investigated using the Hamming distance. In particular, the Hamming distance defined on the set of permutations of a given length was applied to study the fitness landscape of the quadratic assignment problem (Merz and Freisleben, 2000). The feasible set of the QAP constitutes a special case of (1) with $d = 2$, and by virtue of (10) a feasible solution of the QAP can be represented in the form $\mu = (\iota, \pi)$, whereby the Hamming distance between two solutions $\mu_i = (\iota, \pi_i)$ and $\mu_j = (\iota, \pi_j)$ is equal to the number of positions in which $\pi_i(k) \neq \pi_j(k)$, $k = 1, \ldots, n$.

It is easy to see that such a definition does not apply in the case of multidimensional assignment problems (3) with d dimensions and n elements per dimension, where a feasible solution is generally an *unordered* collection of n strings of length d. Thus, for hypergraph matching problems the Hamming distance can be defined in terms of the *minimum* number of positions in which two matchings $\mu_i = \{(i_1^{(1)}, \ldots, i_d^{(1)}), \ldots, (i_1^{(n)}, \ldots, i_d^{(n)})\}$ and $\mu_j = \{(j_1^{(1)}, \ldots, j_d^{(1)}), \ldots, (j_1^{(n)}, \ldots, j_d^{(n)})\}$ differ from each other. For example, the Hamming distance between the following two feasible solutions of a $d = 4$ and $n = 3$ multidimensional assignment problem:

$$\mu_1 = \begin{pmatrix} 1111 \\ 2222 \\ 3333 \end{pmatrix}, \quad \mu_2 = \begin{pmatrix} 1222 \\ 2111 \\ 3333 \end{pmatrix},$$

is equal to 2, but not 6. In general, let us define the number of elements by which the k-th hyperedge in μ_i differs from the ℓ-th hyperedge in μ_j as

$$\Delta_{k\ell} = \left\| (i_1^{(k)}, \ldots, i_d^{(k)}) - (j_1^{(\ell)}, \ldots, j_d^{(\ell)}) \right\| = \sum_{r=1}^{d} \bar{\delta}_{i_r^{(k)} j_r^{(\ell)}}, \tag{31}$$

where $\bar{\delta}_{ij}$ is the negation of Kronecker's delta, $\bar{\delta}_{ij} = 1 - \delta_{ij}$. One evidently has $\Delta_{k\ell} = \Delta_{\ell k}$, and $0 \leq \Delta_{k\ell} \leq d$. Then, the Hamming distance $d_H(\cdot\,,\cdot)$ between hypergraph matchings is defined as the

optimal value of the following LAP:

$$d_H(\mu_i, \mu_j) = \|\mu_i - \mu_j\| = \min_{\pi \in \Pi_n} \sum_{k=1}^{n} \Delta_{k,\pi(k)}, \tag{32}$$

where Π_n is the set of all permutations $\pi : [\![n]\!] \mapsto [\![n]\!]$. Below we discuss some properties of the introduced Hamming distance measure $d_H(\cdot, \cdot)$ as an optimal value of the linear assignment problem (32).

3.1 Diameter of the feasible set

Given a distance measure $d(\cdot, \cdot)$ in a combinatorial optimization problem, the *diameter* of the feasible set can be defined as the maximum distance between two feasible solutions. The Hamming diameter D of the multidimensional assignment problem (3) is then defined as the maximum value of the Hamming distance (32):

$$D = \max_{\mu_i, \mu_j \in \mathcal{M}(\mathcal{H}_{d|n})} d_H(\mu_i, \mu_j). \tag{33}$$

Then, we have the following simple observation:

Proposition 1. *The Hamming diameter D of the feasible set of a multidimensional assignment problem* (3) *of dimensionality d and cardinality n satisfies*

$$D \leq n(d-1), \tag{34}$$

with the equality attained for problems with $n \geq d$.

Proof. Inequality (34) for D follows trivially from the permutation representation (10) of a feasible solution of the MAP (3). To show that this bound is exact for problems with $n \geq d$, consider without loss of generality the distance between the trivial solution $\mu_1 = \{(1, \ldots, 1), (2, \ldots, 2), \ldots, (n, \ldots, n)\}$ given by identity permutations $\pi_1, \ldots, \pi_{d-1} = \iota$ in (10), and a solution $\hat{\mu} = (\hat{\pi}_1, \ldots, \hat{\pi}_d)$, where $\pi_1 = \iota$, and $\hat{\pi}_k = \sigma(\hat{\pi}_{k-1})$, with σ being the forward cyclical permutation. Since there are $n \geq d$ different cyclical permutations of $[\![n]\!]$ (including the identity permutation), we have that for the feasible solutions μ_1 and $\hat{\mu}$ the quantities $\Delta_{k\ell}$ defined by (31) satisfty

$$\Delta_{k\ell} = \sum_{r=1}^{d} \bar{\delta}_{k,\hat{\pi}_r(\ell)} = d - 1,$$

whence $d_H(\mu_1, \hat{\mu}) = n(d-1)$. ∎

3.2 Expected distance to global optimum

Many heuristic solution algorithms for combinatorial problems rely, at least partly, on local search. In the context of assignment problems, local search is often conducted by permuting several (usually, two) positions in the current solution. In this respect, it is of interest to estimate the number of permutations that are necessary to reach a global optimum from a current feasible solution, or, in other words, the

Hamming distance from the given solution to the optimal one. To this end, it is convenient to assume that the matching cost function $\Phi(\mu)$ in (3) has the form

$$\Phi(\mu) = \coprod_{e_{i_1},\ldots,e_{im}\in\mu} \phi_{e_{i_1}\cdots e_{im}}, \tag{35}$$

where $\coprod$ is an operator defined over some set of cost elements $\{\phi\}$ indexed by the hyperedges of $\mathcal{H}_{d|n}$. If the cost elements ϕ are independent identically distributed (iid) random variables with a continuous distribution, then problem (3) has a unique global minimum (since the costs of all feasible solutions are different almost surely), and the location of the optimal matching is uniformly distributed over the feasible set.

Let H denote the expected Hamming distance between the (unique) global minimum and a given feasible solution of a random MAP; then the expected value of H can be computed as

$$\mathsf{E}[H] = \mathsf{E}\big[\mathsf{E}[H \mid Y]\big] = \sum_{k=1}^{n!^{d-1}} \mathsf{E}[H \mid Y]\,\mathsf{P}\{Y = k\}, \tag{36}$$

where the random variable Y takes value k ($1 \le k \le n!^{d-1}$) if k-th feasible solution is the global minimum, and zero otherwise, and $\mathsf{E}[H \mid Y]$ is the conditional expectation of Hamming distance to the global optimum given its location. Since $0 \le H \le n(d-1)$ per Proposition 1, one has

$$\mathsf{E}[H \mid Y] = \sum_{p=0}^{n(d-1)} p\,\mathsf{P}\{H = p \mid Y\} = \sum_{p=0}^{n(d-1)} p\,\frac{N_p}{n!^{d-1}}, \tag{37}$$

where N_p is the number of feasible solutions located at the Hamming distance p from the given solution. If the assignment costs $\{\phi\}$ in (35) are iid continuous, then

$$\mathsf{E}[H] = \sum_{p=0}^{n(d-1)} p\,\frac{N_p}{n!^{d-1}} \tag{38}$$

Again, given that the location of the global minimum is uniformly distributed, the expression for $\mathsf{E}[H]$ above can be interpreted as the *expected diameter* of the feasible set of (3). The expected Hamming distance $\mathsf{E}[H]$ for any distribution of assignment costs can be computed explicitly in the special case $n = 2$.

Proposition 2. *Consider a hypergraph matching problem* (3) *with cardinality parameter $n = 2$, and cost function* (35) *where the cost elements $\{\phi\}$ are iid random variables with a continuous distribution. Then, the expected Hamming distance $\mathsf{E}[H]$ from a given solution to the global optimum is equal to*

$$\mathsf{E}[H] = \frac{1}{2^{d-1}} \left\{ \sum_{s=1}^{\lfloor \frac{d}{2} \rfloor - 1} 2s \binom{d}{s} + 2^{d \bmod 2} \left\lfloor \frac{d}{2} \right\rfloor \binom{d}{\lfloor d/2 \rfloor} \right\}, \tag{39}$$

where $d \bmod 2$ is the remainder on division of d by 2.

Proof. Since in all dimensions of an $n = 2$ MAP there are only two elements, then Hamming distance between two feasible solutions is always even: $d_H(\mu_i, \mu_j) = 0, 2, 4, \ldots$, and equals to the number of dimensions in which the two solutions differ. It is easy to see that permuting elements in all d dimensions of a solution leaves it unchanged; similarly, making $d - 1$ permutations in dimensions $2, 3, \ldots, d$ is the same as permuting dimension 1, and so on. Hence, the Hamming distance between two solutions of an $n = 2$ MAP may take values

$$H = 2s, \quad s = 0, 1, \ldots, \left\lfloor \frac{d}{2} \right\rfloor,$$

where s is the number of dimensions permuted. The number N_{2s} of feasible solutions that are at a distance $H = 2s$ from the given solution is given by $\binom{d}{s}$ for $s = 1, \ldots, \lfloor d/2 \rfloor - 1$. A special case arises when d is even and $s = \lfloor d/2 \rfloor = d/2$, where the number of solutions that are at a distance $H = 2s = d$ from a given solution is equal to $\frac{1}{2}\binom{d}{s}$ due to symmetry; for instance, $\begin{pmatrix} 1212 \\ 2121 \end{pmatrix}$ and $\begin{pmatrix} 2121 \\ 1212 \end{pmatrix}$ represent the same solution. Observe that this does not occur when $s = \lfloor d/2 \rfloor$ and d is odd. ■

In the general case, computation of the number N_p of feasible solutions that are at a given distance p from a specified solution presents significant challenges due to the combinatorial nature of its definition via a solution of the LAP (32). An upper bound on N_p can be obtained by ignoring these combinatorial considerations.

Proposition 3. *In a hypergraph matching problem* (3) *with $d \leq n$, the number N_p of feasible solutions located at a distance p, $2 \leq p \leq n(d-1)$, from a given solution, is bounded from above as*

$$N_p \leq \sum_{(p_1,\ldots,p_k)} \binom{d}{k} \prod_{i=1}^{k} \binom{n}{p_i} D(p_i), \quad \textit{where} \quad D(k) = \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} j!, \tag{40}$$

and the summation is over all vectors $(p_1, \ldots, p_k)$ such that $\sum_{i=1}^{k} p_i = p$, $0 \leq p_i \leq n$, $p_i \neq 1$.

Proof. Expression (40) is obtained by selecting k out of d dimensions in representation (9) of a feasible solution of problem (3), and permuting p_i out of n elements in each dimension, where the $(p_1, \ldots, p_k)$ satisfy the above conditions. To make sure that representation matix of the permuted solution has the largest possible number of entries different from the corresponding elements of the original matrix, we count only those permutations that do not leave any of the p_i permuted elements unchanged. The number of such permutations, or *derangements,* is given by $D(p_i)$ in (40) (see, e.g., Stanley, 1986). ■

Since bound (40) ignores the combinatorial structure imposed by (32), it is reasonable to expect that it can be rather loose, especially for larger values of p, when the combinatorial effects will be most prominent. However, for smaller values of p, pertinent to solution algorithms, the combinatorial effects of definition (32) should be lessened, making the bound (40) tighter. These conclusions are supported by the numerical studies of Hamming distance in MAPs that are presented next.

3.3 Numerical results

In this section we report the results of computational experiments conducted on determining the expected Hamming distance to the global optimum (or, equivalently, to the given feasible solution) $\mathsf{E}[H]$, and the distribution of the numbers N_p of feasible solutions located at a Hamming distance p from the given

solution. The computational studies involved exhaustive enumeration of the feasible set of a multidimensional assignment problem (3) and computing the Hamming distance (32) between each pair of feasible solutions using the shortest augmenting path algorithm for dense LAPs (Jonker and Volgenant, 1987). Due to the enumerative nature of this case study, only small size MAPs ($d = 3, \ldots, 8$, $n = 2, \ldots, 5$) were examined.

Figure 1 displays the expected Hamming distance $\mathsf{E}[H]$ to the global minimum solution of the MAP (3). Interestingly, $\mathsf{E}[H]$ exhibits practically linear growth when one of the parameters d or n is fixed and the other increases. This observation supports our interpretation of $\mathsf{E}[H]$ as the *expected diameter* of the feasible set of (3).

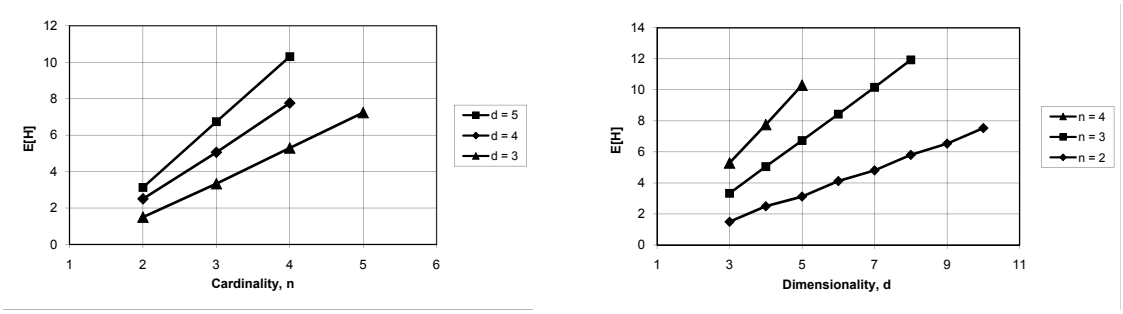


Figure 1: Expected Hamming distance from a given solution (global optimum) in multidimensional assignment problem (3) for fixed dimensionality d or cardinality n.

Figure 2 displays typical distributions of the number N_p of feasible solutions that are located exactly at a Hamming distance p from the given solution. In particular, the distributions of N_p are generally skewed towards larger values of p. Table 1 presents a comparison of the numerically determined values of N_p and the corresponding upper bound as given by Proposition (3). As expected, the upper bound (40) is very loose for larger values of the Hamming distance p, when the combinatorial properties of definition (32) are dominant, and is quite tight for smaller values of p, when the combinatorial effects of (32) are less pronounced.

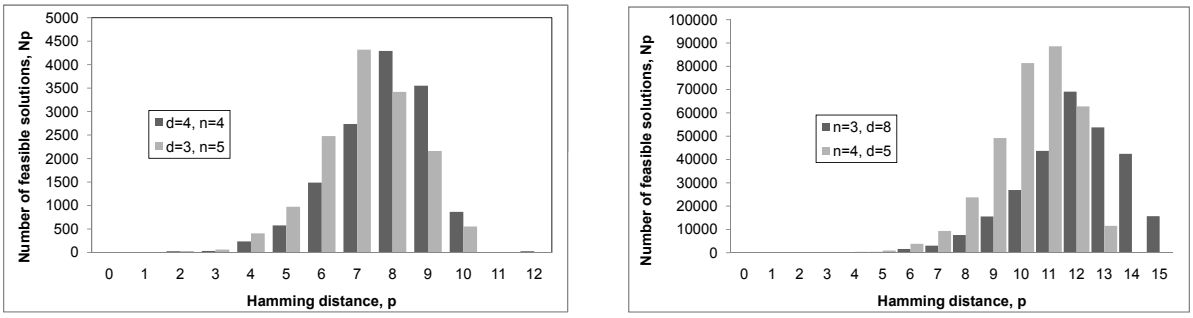


Figure 2: Number N_p of feasible solutions of a multidimensional assignment problem (3) located at a Hamming distance p from a given solution (global optimum).

Table 1: Illustration of the tightness of the upper bound (40) for the number N_p of feasible solutions of MAP ($d = 4, n = 4$) located at a distance p from the given solution.

p	2	3	4	5	6	7	8	9	10	11	12
Bound	24	32	252	576	1896	4320	6390	12416	12744	7776	729
N_p	24	32	234	576	1488	2736	4293	3552	864	0	24

4 High-quality Solution Sets in Randomized Multidimensional Assignment Problems

In this section two methods will be described that can be used to obtain mathematically proven high-quality solutions for MAPs with large cardinality, or large dimensionality. These methods utilize the concept of *index graph* of the underlying hypergraph of the problem.

4.1 Random Linear MAPs of Large Cardinality

In the case when the cost Φ of hypergraph matching is a linear function of hyperedges' costs, i.e., for MAPs with linear objectives, a useful tool for constructing high quality solutions for instances with large cardinality ($n \gg 1$) is the so-called *index graph*. The index graph is related to the concept of *line graph*, in that the vertices of the index graph represent the hyperedges of the hypergraph.

Namely, by indexing each vertex of the index graph $\mathcal{G}^* = (\mathcal{V}^*, \mathcal{E}^*)$ by $(i_1, \ldots, i_d) \in \{1, \ldots, n\}^d$, identically to the corresponding hyperedge of $\mathcal{H}_{d|n}$, the set of vertices $\mathcal{V}^*$ can be partitioned into n subsets $\mathcal{V}^*_k$, also called *levels*, which contain vertices whose first index is equal to k:

$$\mathcal{V}^* = \bigcup_{k=1}^{n} \mathcal{V}^*_k, \quad \mathcal{V}^*_k = \{(k, i_2, \ldots, i_d) \,\big|\, i_2, \ldots, i_d \in \{1, \ldots, n\}\}.$$

For any two vertices $i, j \in \mathcal{V}^*$, an edge (i, j) exists in $\mathcal{G}^*$, $(i, j) \in \mathcal{E}^*$, if and only if the corresponding hyperedges of $\mathcal{H}_{d|n}$ do not have common nodes. In other words,

$$\mathcal{E}^* = \{(i, j) \,\big|\, i = (i_1, \ldots, i_d), j = (j_1, \ldots, i_d) : \; i_k \neq j_k, \; k = 1, \ldots, n\}.$$

Then, that it is easy to see that $\mathcal{G}^*$ has the following properties.

Lemma 2. *Consider a complete, d-partite, n-uniform hypergraph $\mathcal{H}_{d|n} = (\mathcal{V}, \mathcal{E})$, where $|\mathcal{E}| = n^d$, and $\mathcal{V} = \bigcup_{k=1}^{d} \mathcal{V}_k$ such that $\mathcal{V}_k \cap \mathcal{V}_l = \emptyset$, $k \neq l$ and $|\mathcal{V}_k| = n$, $k = 1, \ldots, d$. Then, the index graph $\mathcal{G}^* = (\mathcal{V}^*, \mathcal{E}^*)$ of $\mathcal{H}_{d|n}$ satisfies:*

1. *$\mathcal{G}^*$ is n-partite, namely $\mathcal{V}^* = \bigcup_{k=1}^{n} \mathcal{V}^*_k$, $\mathcal{V}^*_i \cap \mathcal{V}^*_j = \emptyset$ for $i \neq j$, where each $\mathcal{V}^*_k$ is an independent set in $\mathcal{V}^*$: for any $i, j \in \mathcal{V}^*_k$ one has $(i, j) \notin \mathcal{E}^*$*
2. *$|\mathcal{V}^*_k| = n^{d-1}$ for each $k = 1, \ldots, n$*
3. *The set of perfect matchings in $\mathcal{H}_{d|n}$ is isomorphic to the set of n-cliques in $\mathcal{G}^*$, i.e., each perfect matching in $\mathcal{H}_{d|n}$ corresponds uniquely to a (maximum) clique of size n in $\mathcal{G}^*$.*

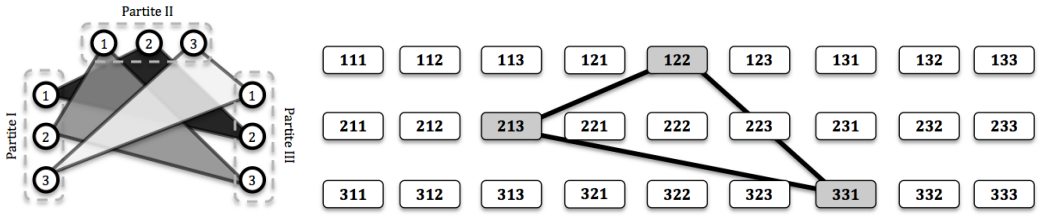


Figure 3: A 3-uniform 3-partite hypergraph $\mathcal{H}_{d|n}$ and its index graph $\mathcal{G}^*$. The vertices of $\mathcal{G}^*$ shaded in grey represent a clique (or, equivalently, a perfect matching on $\mathcal{H}_{d|n}$).

Let us denote by $\mathcal{G}^*(\alpha_n)$ the induced subgraph of the index graph $\mathcal{G}^*$ obtained by *randomly* selecting α_n vertices from each level $\mathcal{V}_k^*$ of $\mathcal{G}^*$, and also define $N(\alpha_n)$ to be the number of cliques in $\mathcal{G}^*(\alpha_n)$, then based on the following lemma Krokhmal et al. (2007) one can select α_n in such a way that $\mathcal{G}^*(\alpha_n)$ is expected to contain at least one n-clique:

Lemma 3. *The subgraph $\mathcal{G}^*(\alpha_n)$ is expected to contain at least one n-clique, or a perfect matching on $\mathcal{H}_{d|n}$ (i.e., $E[N(\alpha_n)] \geq 1$) when α_n is equal to*

$$\alpha_n = \left\lceil \frac{n^{d-1}}{n!^{\frac{d-1}{n}}} \right\rceil. \tag{41}$$

In the case when the cost coefficients $\phi_{i_1 \cdots i_d}$ of MAP with linear or bottleneck objective are drawn independently from a given probability distribution, Lemma 3 can be used to construct high quality solutions. The approach is to create the subgraph $\mathcal{G}^*_{\min}(\alpha_n)$, also called the *$\alpha$-set*, from the index graph $\mathcal{G}^*$ of the MAP by selecting α_n nodes with the smallest cost coefficients from each partition (level) of $\mathcal{G}^*$. If the costs of the hyperedges of $\mathcal{H}_{d|n}$, or, equivalently, vertices of $\mathcal{G}^*$, are identically and independently distributed, the α-set is expected to contain at least one clique, which represents a perfect matching in the hypergraph $\mathcal{H}_{d|n}$. It should be noted that since the α-set is created from the nodes with the smallest cost coefficients, if a clique exists in the α-set, the resulting cost of the perfect matching is expected to be close to the optimal solution of the MAP.

Importantly, when the cardinality n of the MAP increases, the size of the subgraph $\mathcal{G}^*(\alpha_n)$ or $\mathcal{G}^*_{\min}(\alpha_n)$ grows only as $O(n)$, as evidenced by the following observation:

Lemma 4. *If d is fixed and $n \to \infty$, then α_n monotonically approaches a finite limit:*

$$\alpha_n \nearrow \alpha := \lceil e^{d-1} \rceil \quad as \quad n \nearrow \infty. \tag{42}$$

Corollary 3.1. *In the case of randomized MAP of large enough cardinality $n \gg 1$ the subset $\mathcal{G}^*_{\min}$ expected to contain a high-quality feasible solution of the MAP can simply be chosen as $\mathcal{G}^*_{\min}(\alpha)$, where α is given by* (42)*.*

Observe that using the α-set $\mathcal{G}^*_{\min}(\alpha)$ for construction of a low-cost feasible solution to randomized MAP with linear or bottleneck objectives may prove to be a challenging task, since it is equivalent to finding an n-clique in an n-partite graph; moreover, the graph $\mathcal{G}^*_{\min}(\alpha)$ is only expected to contain a single n-clique (feasible solution). The following variation of Lemma 3 allows for constructing a subgraph of $\mathcal{G}^*$ that contains exponentially many feasible solutions:

Lemma 5. *Consider the index graph $\mathcal{G}^*$ of the underlying hypergraph $\mathcal{H}_{d|n}$ of a randomized MAP, and let*

$$\beta_n = \left\lceil 2\frac{n^{d-1}}{n!^{\frac{d-1}{n}}} \right\rceil. \tag{43}$$

Then, the subgraph $\mathcal{G}^*(\beta_n)$ is expected to contain 2^n n-cliques, or, equivalently, perfect matching on $\mathcal{H}_{d|n}$.

Proof. The statement of the lemma is easy to obtain by regarding the feasible solutions of the MAP as *paths* that contain exactly one vertex in each of the n "levels" $\mathcal{V}_1^*, \ldots, \mathcal{V}_n^*$ of the index graph $\mathcal{G}^*$. Namely, let us call a path connecting the vertices $(1, i_2^{(1)}, \ldots, i_d^{(1)}) \in \mathcal{V}_1^*$, $(2, i_2^{(2)}, \ldots, i_d^{(2)}) \in \mathcal{V}_2^*$, $\ldots$, $(n, i_2^{(n)}, \ldots, i_d^{(n)}) \in \mathcal{V}_n^*$ *feasible* if $\{i_k^{(1)}, i_k^{(2)}, \ldots, i_k^{(n)}\}$ is a permutation of the set $\{1, \ldots, n\}$ for every $k = 2, \ldots, d$. Note that from the definition of the index graph $\mathcal{G}^*$ it follows that a path is feasible if and only if the vertices it connects form an n-clique in $\mathcal{G}^*$. Next, observe that a path in $\mathcal{G}^*$ chosen at random is feasible with the probability $\left(\frac{n!}{n^n}\right)^{d-1}$, since one can construct $n^{n(d-1)}$ different (not necessarily feasible) paths in $\mathcal{G}^*$. Then, if we randomly select β_n vertices from each set $\mathcal{V}_k^*$ in such a way that out of the $(\beta_n)^n$ paths spanned by $\mathcal{G}^*(\beta_n)$ at least 2^n are feasible, the value of β_n must satisfy:

$$(\beta_n)^n \left(\frac{n!}{n^n}\right)^{d-1} \geq 2^n,$$

from which it follows immediately that β_n must satisfy (43). ■

Corollary 3.2. *If d is fixed and $n \to \infty$, then β_n monotonically approaches a finite limit:*

$$\beta_n \nearrow \beta := \lceil 2e^{d-1} \rceil \quad as \quad n \nearrow \infty.$$

Remark 1. *Since the value of the parameter β_n (43) is close to the double of the parameter α_n (41), the subgraph $\mathcal{G}^*_{\min}(\beta_n)$, constructed from selecting β_n nodes with the smallest cost coefficients from each partition (level) of $\mathcal{G}^*$ will be called the "2α-set", or $\mathcal{G}^*(2\alpha)$.*

Following Krokhmal and Pardalos (2011), the costs of feasible solutions of randomized MAPs with linear or bottleneck objectives that are contained in the α- or 2α-sets can be shown to satisfy:

Lemma 6. *Consider a randomized MAP with linear or bottleneck objectives, whose cost coefficients are iid random variables from a continuous distribution F with a finite left endpoint of the support, $F^{-1}(0) > -\infty$. Then, for a fixed $d \geq 3$ and large enough values of n, if the subset $\mathcal{G}^*_{\min}(\alpha)$ (or, respectively, $\mathcal{G}^*_{\min}(\beta)$) contains a feasible solution of the MAP, the cost Z_n of this solution satisfies*

$$(n-1)F^{-1}(0) + F^{-1}\left(\frac{1}{n^{d-1}}\right) \leq Z_n \leq nF^{-1}\left(\frac{3\ln n}{n^{d-1}}\right), \quad n \gg 1, \tag{44}$$

in the case of MAP with linear objective (5)*, while in the case of MAP with bottleneck objective* (7) *the cost W_n of such a solution satisfies*

$$F^{-1}\left(\frac{1}{n^{d-1}}\right) \leq W_n \leq F^{-1}\left(\frac{3\ln n}{n^{d-1}}\right), \quad n \gg 1. \tag{45}$$

4.2 Random MAPs of Large Dimensionality

In cases where the cardinality of the MAP is fixed, and its dimensionality is large, $d \gg 1$, the approach described in section 4.1 based on the construction of α- or 2α-subset of the index graph $\mathcal{G}^*$ of the MAP is not well suited, since in this case the size of $\mathcal{G}^*(\alpha)$ grows exponentially in d.

However, the index graph $\mathcal{G}^*$ of the underlying hypergraph $\mathcal{H}_{d|n}$ of the MAP can still be utilized to construct high-quality solutions of large-dimensionality randomized MAPs.

Let us call two matchings $\mu_i = \{(i_1^{(1)}, \ldots, i_d^{(1)}), \ldots, (i_1^{(n)}, \ldots, i_d^{(n)})\}$ and $\mu_j = \{(j_1^{(1)}, \ldots, j_d^{(1)}), \ldots, (j_1^{(n)}, \ldots, j_d^{(n)})\}$ on the hypergraph $\mathcal{H}_{d|n}$ *disjoint* if

$$(i_1^{(k)}, \ldots, i_d^{(k)}) \neq (j_1^{(\ell)}, \ldots, j_d^{(\ell)}) \quad \text{for all} \quad 1 \leq k, \ell \leq n,$$

or, in other words, if μ_i and μ_j do not have any common hyperedges. It is easy to see that if the cost coefficients of randomized MAPs are iid random variables, then the costs of the feasible solutions corresponding to the disjoint matchings are also independent and identically distributed.

Next, we show how the index graph $\mathcal{G}^*$ of the MAP can be used to construct exactly n^{d-1} disjoint solutions whose costs are iid random variables. First, recalling the interpretation of feasible MAP solutions as *paths* in the index graph $\mathcal{G}^*$, we observe that disjoint solutions of MAP, or, equivalently, disjoint matchings on $\mathcal{H}_{d|n}$ are represented by disjoint paths in $\mathcal{G}^*$ that do not have common vertices.

Note that since each level $\mathcal{V}_k^*$ of $\mathcal{G}^*$ contains exactly n^{d-1} vertices (see Lemma 2), there may be no set of disjoint paths with more than n^{d-1} elements.

On the other hand, recall that a (feasible) path $\mathcal{G}^*$ can be described as a set of n vectors

$$\mu = \{(i_1^{(1)}, \ldots, i_d^{(1)}), \ldots, (i_1^{(n)}, \ldots, i_d^{(n)})\},$$

such that $\{i_k^{(1)}, \ldots, i_k^{(n)}\}$ is a permutation of the set $\{1, \ldots, n\}$ for each $k = 1, \ldots, d$. Then, for any given vertex $v^{(1)} = (1, i_2^{(1)}, \ldots, i_d^{(1)}) \in \mathcal{V}_1^*$, let us construct a feasible path containing $v^{(1)}$ in the form

$$\{(1, i_2^{(1)}, \ldots, i_d^{(1)}), (2, i_2^{(2)}, \ldots, i_d^{(2)}), \ldots, (n, i_2^{(n)} \cdots i_d^{(n)})\},$$

where for $k = 2, \ldots, d$ and $r = 2, \ldots, n$

$$i_k^{(r)} = \begin{cases} i_k^{(r-1)} + 1, & \text{if } i_k^{(r-1)} = 1, \ldots, n-1, \\ 1, & \text{if } i_k^{(r-1)} = n. \end{cases} \tag{46}$$

In other words, $\{i_k^{(1)}, \ldots, i_k^{(n)}\}$ is a forward cyclic permutation of the set $\{1, \ldots, n\}$ for any $k = 2, \ldots, d$. Applying (46) to each of the n^{d-1} vertices $(1, i_2^{(1)}, \ldots, i_d^{(1)}) \in \mathcal{V}_1^*$, we obtain n^{d-1} feasible paths (matchings on $\mathcal{H}_{d|n}$) that are mutually disjoint, since (46) defines a bijective mapping between any vertex (hyperedge) $(k, i_2^{(k)}, \ldots, i_d^{(k)})$ from the set $\mathcal{V}_k^*$, $k = 2, \ldots, n$, and the corresponding vertex (hyperedge) $v^{(1)} \in \mathcal{V}_1^*$.

Then, if hyperedge costs $\phi_{i_1 \cdots i_d}$ in the linear or bottleneck MAPs (5) and (7) are stochastically independent, the costs $\Phi(\mu_1), \ldots, \Phi(\mu_{n^{d-1}})$ of the n^{d-1} disjoint matchings $\mu_1, \ldots, \mu_{n^{d-1}}$ defined by (46) are also independent, as they do not contain any common elements $\phi_{i_1 \cdots i_d}$. Given that the optimal solution cost $Z_{d,n}^*$ (respectively, $W_{d,n}^*$) of randomized linear (respectively, bottleneck) MAP does not exceed the costs $\Phi(\mu_1), \ldots, \Phi(\mu_{n^{d-1}})$ of the disjoint solutions described by (46), the following bound on the optimal cost of linear or bottleneck randomized MAP can be established:

Lemma 7. *The optimal costs $Z^*_{d,n}$, $W^*_{d,n}$ of random MAPs with linear or bottleneck objectives* (5), (7)*, where cost coefficients are iid random variables, satisfy*

$$Z^*_{d,n} \leq X^{\sum}_{1:n^{d-1}}, \quad W^*_{d,n} \leq X^{\max}_{1:n^{d-1}}, \tag{47}$$

where $X^{\sum}_i$, $X^{\max}_i$ $(i = 1, \ldots, n^{d-1})$ are iid random variables with distributions $F^{\sum,\max}$ that are determined by the form of the corresponding objective function, and $X_{1:k}$ denotes the minimum-order statistic among k iid random variables.

Remark 2. *Inequalities in* (47) *are tight: namely, in the special case of random MAPs with $n = 2$, all of the $n!^{d-1} = 2^{d-1}$ feasible solutions are stochastically independent Grundel et al. (2007), whereby equalities hold in* (47)*.*

As shown in Krokhmal and Pardalos (2011), the following quality guarantee on the minimum cost of the n^{d-1} disjoint solutions (46) of linear and bottleneck MAPs can be established:

$$X^{\sum}_{1:n^{d-1}} \leq nF^{-1}\left(n^{-\frac{d-1}{2n}}\right), \quad X^{\max}_{1:n^{d-1}} \leq F^{-1}\left(n^{-\frac{d-1}{2n}}\right), \quad d \gg 1,$$

where F^{-1} is the inverse of the distribution function F of the cost coefficients $\phi_{i_1 \cdots i_d}$. This observation allows for constructing high-quality solutions of randomized linear and bottleneck MAPs by searching the set of disjoint feasible solutions as defined by (46).

4.3 Numerical Results

Sections 4.1 and 4.2 introduced two methods of solving randomized instances of MAPs by constructing subsets (neighborhoods) of the feasible set of the problem that are guaranteed to contain high-quality solutions whose costs approach optimality when the problem size ($n \to \infty$, or, respectively, $d \to \infty$) increases. In this section we investigate the quality of solutions contained in these neighborhoods for small- to moderate-sized problem instances, and compare the results with the optimal solutions where it is possible.

Before proceeding with the numerical results of the study, in the next section, FINDCLIQUE, the algorithm that is used to find the optimum clique in the index-graph $\mathcal{G}^*$ or the first clique in the α-set or 2α-set will be described. The results from randomly generated MAP instances for each of these two methods are presented next.

4.3.1 Finding n-Cliques in n-Partite Graphs

In order to find cliques in $\mathcal{G}^*$, the α-set, or the 2α-set, the branch-and-bound algorithm proposed in **?** is used. This algorithm, called FINDCLIQUE, is designed to find all n-cliques contained in an unweighted n-partite graph.

The input to original FINDCLIQUE is an n-partite graph $G(V_1, \ldots, V_n; E)$ with the adjacency matrix $\mathsf{M} = (m_{ij})$, and the output will be a list of all n-cliques contained in G. Nodes from G are copied into a set called `compatible nodes`, denoted by C. The set C is further divided into n partitions, each denoted by C_i that are initialized such that they contain nodes from partite V_i, $i = \{1, \ldots, n\}$. FINDCLIQUE also maintains two other sets, namely, `current clique`, denoted by Q and `erased`

`nodes`, denoted by E. The set Q holds a set of nodes that are pairwise adjacent and construct a clique. The `erased node` set, E, is furthered partitioned into n sets, denoted by E_i, that are initialized as empty. At each step of the algorithm, E_i will contain the nodes that are not adjacent to the i^{th} node added to Q.

The branch-and-bound tree has n levels, and FINDCLIQUE searches for n-cliques in the tree in a depth-first fashion. At level t of the branch of bound algorithm, the index of the smallest partition in C, $\theta = \arg\min_i\{|C_i| \,\big|\, i \notin \mathsf{V}\}$ will be detected, and C_θ will be marked as visited by including θ into $\mathsf{V} \leftarrow \{\mathsf{V} \cup \theta\}$, where V is the list of partitions that have a node in Q. Then, a node q from C_θ is selected at random and added to Q. If $|Q| = n$, an n-clique is found. Otherwise, C will be updated; every partition C_i where $i \notin \mathsf{V}$ will be searched for nodes c_{ij}, $(j = 1, \ldots, |C_i|)$ that are not adjacent to q, i.e. $m_{q,c_{ij}} = 0$. Any such node will be removed from C_i and will be transferred to E_t. Note that in contrast to C, nodes in different levels of E will not necessarily be from the same partite of G. Decision regarding backtracking is made after C is updated. It is obvious that in an n-partite graph the following will hold:

$$\omega(G) \leq n, \tag{48}$$

where $\omega(G)$ is the size of a maximum clique in G. In other words, the size of any maximum clique cannot be larger than the number of partites, in that the maximum clique can only contain at most 1 node from each partite of G. If after updating, there is any $C_i \notin \mathsf{V}$ with $|C_i| = 0$, adding q_i to Q will not result in a clique of size n, since the condition in (48) changes into strict inequality. In such cases, q is removed from Q, nodes from E_t will be transferred back to their respective partitions in C, and FINDCLIQUE will try to add another node from C_θ that is not already branched on, to Q. If such a node does not exist, the list of visited partitions will be updated ($\mathsf{V} \leftarrow \mathsf{V} \backslash \theta$), and FINDCLIQUE backtracks to the previous level of the branch-and-bound tree. If the backtracking condition is not met and q is promising, FINDCLIQUE will go one level deeper in the tree, finds the next smallest partition in the updated C and tries to add a new node to Q.

When solving the clique problem in the α-set or 2α-set, since the objective is to find the first n-clique regardless of its cost, FINDCLIQUE can be used without any modifications, and the weights of the nodes in $\mathcal{G}^*_{\min}(\alpha)$ or $\mathcal{G}^*_{\min}(2\alpha)$ will be ignored. However, when the optimal clique with the smallest cost in $\mathcal{G}^*$ is sought, some modifications in FINDCLIQUE are necessary to enable it to deal with weighted graphs. The simplest way to adjust FINDCLIQUE is to compute the weight of the n-cliques as they are found, and report the clique with the smallest cost as the output of the algorithm. This is the method that is used in the experimental studies whenever the optimal solution is desired. However, to obtain a more efficient algorithm, it is possible to calculate the weight of the partial clique contained in Q in every step of the algorithm and fathom subproblems for which $W_Q \geq W_{Q^*}$, where W_Q and W_{Q^*} are the cost of the partial clique in Q and the cost of the best clique found so far by the algorithm respectively. Further improvement can be achieved by sorting the nodes in C_i, $i = 1, \ldots, n$, based on their cost coefficients, and each time select the untraversed node with the smallest node as the next node to be added to Q (as opposed to randomly selecting a node, which does not change the overall computational time in the unweighted graph if a list of all n-cliques is desired). This enables us to compute a lower bound on the cost of the maximum clique that the nodes in Q may lead to as follows:

$$LB_Q = W_Q + \sum_{i \notin \mathsf{V}} w_i^{\min}, \tag{49}$$

where $w_i^{\min}$ is the weight of the node with the smallest cost coefficient in C_i. Any subproblem with $LB_Q \geq W_{Q^*}$ will be fathomed.

4.3.2 Random Linear MAPs of Large Cardinality

To demonstrate the performance of the method described in section 4.1, random MAPs with fixed dimensionality $d = 3$ and different values of cardinality n are generated. The cost coefficients $\phi_{i_1 \cdots i_d}$ are randomly drawn from the uniform $U[0, 1]$ distribution. Three sets of problems are solved for this case: (i) $n = 3, \ldots, 8$ with $d = 3$, solved for optimality, and the first clique in the α- and 2α-sets, (ii) $n = 10, 15, \ldots, 45$, with $d = 3$, solved for the first clique in the α- and 2α-sets, and finally (iii) $n = 50, 55, \ldots, 80$, with $d = 3$, solved for the first clique in the 2α-set. For each value of n, 25 instances are generated and solved by modified FINDCLIQUE for the optimum clique or FINDCLIQUE whenever the first clique in the problem is desired. Algorithm is terminated if the computational time needed to solve an instance exceeds 1 hour.

In the first group, (i), instances of MAP that admit solution to optimality in a reasonable time were solved. The results from this subset are used to determine the applicability of Corollary 3.1 and bounds (44) and (45) for relatively small values of n. Table 2 summarizes the average values for the cost of the clique and computational time needed for MAPs with the linear sum objective function for the instances in group (i). The first column, n, is the cardinality of the problem. The columns under the heading "Exact" contain the values related to the optimal clique in $\mathcal{G}^*$. The columns under the heading "$\mathcal{G}^*_{\min}(\alpha_n)$" represent the values obtained from solving the α-set for the first clique, and those under the heading "$\mathcal{G}^*_{\min}(2\alpha)$" represent the values obtained from solving the 2α-set for the first clique. For each of these multicolumns, T denotes the average computational time in seconds, Z is the average cost of the cliques, $|V|$ is the order of the graph or induced subgraph in $\mathcal{G}^*$, $\mathcal{G}^*_{\min}(\alpha)$, or $\mathcal{G}^*_{\min}(2\alpha)$, and $\exists$ CLQ shows the percentage of the problems for which the α-set or 2α-set, respectively, contains a clique. This value is 100% for the exact method. There was no instances in group (i) for which the computational time exceeded 1 hour.

It is clear that using α-set or 2α-set enables us to obtain a high-quality solution in a much shorter time by merely searching a significantly smaller part of the index graph $\mathcal{G}^*$. Based on the values for Z, the cost of the clique found in α-set or 2α-set are consistently converging to that of the optimal clique and they provide tight upper bounds for the optimum cost. Additionally, as is shown in the $|V|$ column, significant reduction in the size of the graph can be obtained if α-set or 2α-set are used.

Table 3 contains the corresponding results for the case of a random MAP with bottleneck objective. In this table, W represents the value for the cost of the optimal clique or the first clique found in α- or 2α-set. Figure 4(a) shows how the cost of an optimum clique compares to the cost of the clique found in α-set and 2α-set. Clearly, the cost of optimal clique approaches 0 for both linear sum and linear bottleneck MAPs. Figure 4(b) demonstrates the computational time for instances in group (i).

The advantage of using α-set over 2α-set is that the quality of the detected clique is expected to be higher. On average, however, a clique in 2α-set is found in a shorter time than in α-set.

The second group of problems, (ii), comprises instances that cannot be solved to optimality within 1 hour. The range of n for this group is such that the first clique in the α-set is expected to be found within 1 hour. Tables 4 and 5 summarize the results obtained for this group. Instances with $n = 45$ were the largest problems in this group for which α-set could be solved within 1 hour. As it is expected, the 2α-set can be solved quickly in a matter of seconds where the equivalent problem for α-set requires a significantly longer computational time. However, the quality of the solutions found for α-set is higher than the quality for solutions in 2α-set. Nonetheless, using 2α-set increases the odds of finding a clique, as based on lemma 5, 2α-set is expected to contain an exponential number of cliques. It is obvious from the $\exists$ CLQ column that not all of the instances in α-set contain at least a clique, whereas 100%

Table 2: Comparison of the computational time and cost for the optimum clique and the first clique found in $\mathcal{G}^*(\alpha)$ and $\mathcal{G}^*(2\alpha)$ in random MAPs with linear sum objective functions for instances in group (i).

	Exact				$\mathcal{G}^*_{\min}(\alpha)$				$\mathcal{G}^*_{\min}(2\alpha)$			
n	$T^*_{n,3}$	$Z^*_{n,3}$	$\|V\|$	$\exists$ CLQ	$T_{\mathcal{G}_{min}(\alpha)}$	$Z_{\mathcal{G}_{min}(\alpha)}$	$\|V\|$	$\exists$ CLQ	$T_{\mathcal{G}_{min}(2\alpha)}$	$Z_{\mathcal{G}_{min}(2\alpha)}$	$\|V\|$	$\exists$ CLQ
3	0.02	0.604	3×26	100	0.04	0.609	3×3	76	0.03	0.773	3×6	100
4	0.01	0.458	4×63	100	0.03	0.514	4×4	88	0.03	0.635	4×7	100
5	0.02	0.371	5×124	100	0.04	0.399	5×4	72	0.03	0.571	5×8	100
6	0.31	0.374	6×215	100	0.04	0.452	6×5	92	0.01	0.524	6×9	100
7	14.83	0.329	7×342	100	0.04	0.392	7×5	80	0.05	0.47	7×9	100
8	937.67	0.274	8×511	100	0.05	0.329	8×5	72	0.04	0.478	8×10	100

Table 3: Comparison of the computational time and cost for the optimum clique and the first clique found in $\mathcal{G}^*(\alpha)$ and $\mathcal{G}^*(2\alpha)$ in random MAPs with linear bottleneck objective functions for instances in group (i).

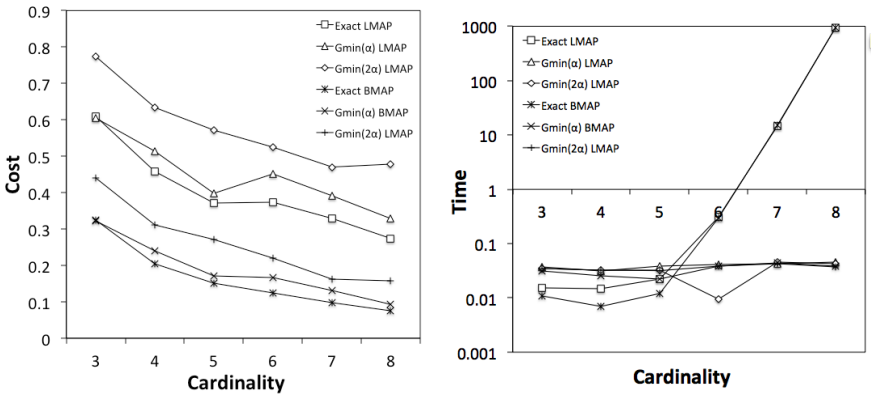
	Exact				$\mathcal{G}^*_{\min}(\alpha)$				$\mathcal{G}^*_{\min}(2\alpha)$			
n	$T^*_{n,3}$	$W^*_{n,3}$	$\|V\|$	$\exists$ CLQ	$T_{\mathcal{G}_{min}(\alpha)}$	$W_{\mathcal{G}_{min}(\alpha)}$	$\|V\|$	$\exists$ CLQ	$T_{\mathcal{G}_{min}(2\alpha)}$	$W_{\mathcal{G}_{min}(2\alpha)}$	$\|V\|$	$\exists$ CLQ
3	0.01	0.321	3×26	100	0.03	0.324	3×3	76	0.04	0.439	3×6	100
4	0.01	0.205	4×63	100	0.03	0.241	4×4	88	0.03	0.311	4×7	100
5	0.01	0.151	5×124	100	0.02	0.17	5×4	72	0.03	0.27	5×8	100
6	0.3	0.124	6×215	100	0.04	0.166	6×5	92	0.04	0.219	6×9	100
7	14.96	0.098	7×342	100	0.04	0.131	7×5	80	0.04	0.163	7×9	100
8	956.6	0.075	8×511	100	0.04	0.092	8×5	72	0.04	0.157	8×10	100

of the instances in 2α-set contain one that can be found within 1 hour. Column *Timeout* represents the percentage of the problems that could not be solved within the allocated 1 hour time limit. Out of 25 instances solved for $n = 45$, only 4 (16%) could not be solved in 1 hour. Out of the 21 remaining instances, 20 instances contained a clique, and only 1 did not have a clique. The behavior of the average cost values for the problems solved in this group are depicted in Figure 5.

Finally, the third group, (iii), includes instances for which the cardinality of the problem prevents the α-set from being solved within 1 hour. Thus, for this set, only the 2α-set is used. The instances of this group were solved with the parameter values $n = 50, 55, \ldots, 80$ and $d = 3$. Tables 6 and 7 summarize the corresponding results. When the size of the problem $n \geq 55$, some instances of problems become impossible to solve within 1 hour time limit. The average cost for the instances that are solved keeps the usual trend and converges to 0 as n grows. The largest problems attempted to be solved in this group are

Table 4: Comparison of the computational time and cost for the first clique found in $\mathcal{G}^*(\alpha)$ and $\mathcal{G}^*(2\alpha)$ in random MAPs with linear sum objective functions for instances in group (ii).

	$\mathcal{G}^*_{\min}(\alpha)$					$\mathcal{G}^*_{\min}(2\alpha)$				
n	$T_{\mathcal{G}_{min}(\alpha)}$	$Z_{\mathcal{G}_{min}(\alpha)}$	$\|V\|$	$\exists$ CLQ	Timeout	$T_{\mathcal{G}_{min}(2\alpha)}$	$Z_{\mathcal{G}_{min}(2\alpha)}$	$\|V\|$	$\exists$ CLQ	Timeout
10	0.05	0.266	10×5	60	-	0.05	0.37	10×10	100	-
15	0.06	0.228	15×6	76	-	0.06	0.313	15×11	100	-
20	0.08	0.165	20×6	56	-	0.07	0.246	20×12	100	-
25	0.15	0.147	25×7	80	-	0.08	0.2	25×13	100	-
30	0.89	0.134	30×7	92	-	0.09	0.171	30×13	100	-
35	8.54	0.11	35×7	88	-	0.14	0.151	35×13	100	-
40	100.85	0.097	40×7	92	-	0.46	0.131	40×13	100	-
45	405.16	0.085	45×7	80	16	1.09	0.122	45×14	100	-



(a) (b)

Figure 4: Solution costs **(a)** and computational time **(b)** in random MAPs with linear sum and linear bottleneck objective functions for instances in group (i).

Table 5: Comparison of the computational time and cost for the first clique found in $\mathcal{G}^*(\alpha)$ and $\mathcal{G}^*(2\alpha)$ in random MAPs with linear bottleneck objective functions for instances in group (ii).

	$\mathcal{G}^*_{\min}(\alpha)$					$\mathcal{G}^*_{\min}(2\alpha)$				
n	$T_{\mathcal{G}_{min}(\alpha)}$	$W_{\mathcal{G}_{min}(\alpha)}$	$\lvert V\rvert$	$\exists$ CLQ	Timeout	$T_{\mathcal{G}_{min}(2\alpha)}$	$W_{\mathcal{G}_{min}(2\alpha)}$	$\lvert V\rvert$	$\exists$ CLQ	Timeout
10	0.04	0.065	10×5	60	-	0.02	0.098	10×10	100	-
15	0.04	0.037	15×6	76	-	0.02	0.056	15×11	100	-
20	0.05	0.023	20×6	56	-	0.04	0.036	20×12	100	-
25	0.1	0.017	25×7	80	-	0.08	0.025	25×13	100	-
30	0.87	0.012	30×7	92	-	0.1	0.019	30×13	100	-
35	8.53	0.009	35×7	88	-	0.15	0.015	35×13	100	-
40	100.99	0.007	40×7	92	-	0.46	0.011	40×13	100	-
45	403.52	0.006	45×7	80	16	1.09	0.009	45×14	100	-

MAPs with $n = 80$. Out of 25 instances of this size, only 4 could be solved within 1 hour. Figure 5(a) the average values of solution cost and computational time for the instances of both linear sum and linear bottleneck MAPs. Note that as the size of the problem increases, the reduction in the size of problem achieved from using α-set or 2α-set becomes significantly larger. For instance, in MAP with $n = 80$ and $d = 3$, the 2α-set has 80×14 nodes, while the complete index graph will have 80×80^2 nodes.

4.3.3 Random MAPs of Large Dimensionality

The second set of problem instances includes MAPs that are solved by the heuristic method explained in section 4.2. Problems in this set have the cardinality $n = 2, \ldots, 5$ and dimensionality in the range $d = 2, \ldots, \bar{d}_n$, where $\bar{d}_n$ is the largest value for d for which an MAP with cardinality n can be solved within 1 hour using the heuristic method. For each pair of (n, d), 25 instances of MAP with cost coefficients randomly drawn from the uniform $U[0, 1]$ distribution are generated. Generated instances are then solved by the modified FINDCLIQUE for the optimal clique (when possible) and the optimal costs are compared with the costs obtained from the heuristic method. The result of the heuristic method for instances with $n = 2$ is optimal, and the heuristic checks all the 2^{d-1} solutions of the MAP. Thus, using the modified FINDCLIQUE to find the optimum clique is not necessary.

Table 6: Computational time and cost for the first clique found in $\mathcal{G}^*(2\alpha)$ in random MAPs with linear sum objective functions for instances in group (iii).

	$\mathcal{G}^*_{\min}(2\alpha)$				
n	$T_{\mathcal{G}_{min}(2\alpha)}$	$Z_{\mathcal{G}_{min}(2\alpha)}$	$\|V\|$	$\exists$ CLQ	Timeout
50	1.56	0.11	50×14	100	-
55	52.29	0.099	55×14	96	4
60	189.9	0.091	60×14	92	8
65	568.9	0.085	65×14	96	4
70	919.79	0.078	70×14	64	36
75	1556.89	0.075	75×14	40	60
80	1641.26	0.07	80×14	16	84

Table 7: Computational time and cost for the first clique found in $\mathcal{G}^*(2\alpha)$ in random MAPs with linear bottleneck objective functions for instances in group (iii).

	$\mathcal{G}^*_{\min}(2\alpha)$				
n	$T_{\mathcal{G}_{min}(2\alpha)}$	$W_{\mathcal{G}_{min}(2\alpha)}$	$\|V\|$	$\exists$ CLQ	Timeout
50	1.56	0.008	50×14	100	-
55	52.19	0.006	55×14	96	4
60	190.6	0.005	60×14	92	8
65	566.71	0.005	65×14	96	4
70	920.44	0.004	70×14	64	36
75	1552.74	0.004	75×14	40	60
80	1631.89	0.003	80×14	16	84

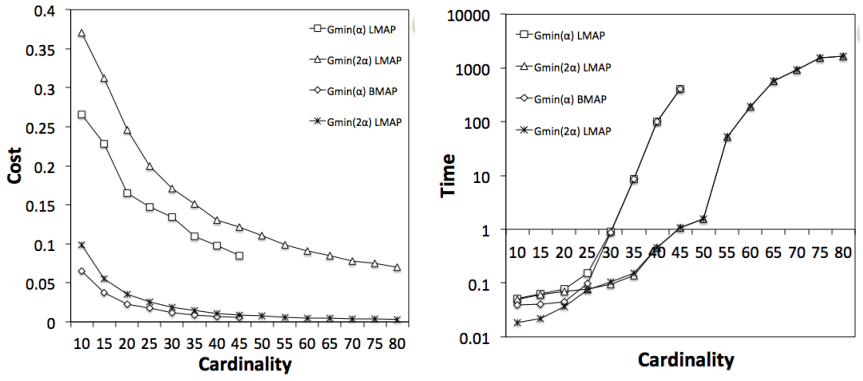
Figure 6 demonstrates the cost convergence in instances with $n = 2, 3, 4, 5$ for both linear sum and linear bottleneck MAPs. Figure 6(a) demonstrates the cost convergence in MAPs with $n = 2$ and $d = 2, \ldots, 27$. Recall that due to Remark 2, for cases with $n = 2$ the heuristic provides the optimal solution. The heuristic method provides high quality solutions that are consistently converging to the optimal solution for all cases and the average value of the obtained costs from the heuristics approaches 0. Memory limitations, as opposed to computational time, were the restrictive factor for solving larger instances as the computational time for the problems of this set never exceeded 700 seconds.

Figure 7 demonstrates the computational time for the optimal method as well as the heuristic method in instances with $n = 2, 3, 4, 5$ for both linear sum and linear bottleneck MAPs. The computational time has an exponential trend as the number of solutions for the MAP, or the number of solutions checked by the heuristic grow in an exponential manner. However, the heuristic method is able to find high quality solutions in significantly shorter time.

5 On γ-cliques in random graphs

A *complete* subset, or a *clique* in a simple undirected graph $G = (V, E)$ is a subset $Q \subseteq V$ of vertices that are pairwise adjacent, i.e., for any $i, j \in Q$ there exists an edge $(i, j) \in E$. A clique Q that cannot be increased by adding new vertices from $V \setminus Q$ is called *maximal*; a clique of the largest size is called the *maximum* clique. In the present endeavor we are concerned with the properties and behavior of a certain type of clique relaxation, namely the *quasi-clique*, or γ*-clique* Abello et al. (2002).

Definition 1 (quasi-clique)**.** *Let $G = (V, E)$ be a simple undirected graph, and $Q \subseteq V$ be a subset of its vertices. The (induced) subgraph $G[Q]$ is called a γ-clique for a given fixed $\gamma \in (0, 1]$, if the ratio of*



(a) (b)

Figure 5: Comparison of the cost **(a)** and computational time **(b)** for MAPs with linear sum and linear bottleneck objective functions for group (ii) and (iii).

the number of edges in $G[Q]$ to the maximum possible number of edges among vertices in Q is at least γ:

$$|E(Q)| \geq \gamma \binom{|Q|}{2}.$$

Note that the case of $\gamma = 0$ would be trivial, as any graph is a 0-clique. A complete graph is a 1-clique, hence γ-clique represents a *density*-based relaxation of the clique, as compared to *degree*– and *diameter*-based clique relaxations such as k-plex and k-club (Balasundaram et al., 2011; Alba, 1973; Mokken, 1979; Wasserman and Faust, 1994; Scott, 2007). Similarly to maximum cliques, a γ-clique with the largest number of vertices is called the *maximum* γ-clique.

In this work, we investigate the asymptotical behavior of γ-cliques in large-scale random graphs, and develop a compact linear mixed-integer programming formulation for identifying the largest γ-clique in a given network.

In particular, we employ a popular $G(n, p)$ model of random graphs, originated by Erdös Erdös (1947), which denotes a graph on n vertices, such that an edge between any two vertices exists with a probability $p \in (0, 1]$, independently from other edges. The $G(n, p)$ model is related to the Erdös–Rényi $G(n, M)$ model of random graphs, in which graphs with n vertices and M edges are uniformly equiprobable with probability $\binom{\binom{n}{2}}{M}^{-1}$. For the $G(n, p)$ model yields graph instances with a rather "uniform" structure, as opposed to, say, *power-law* graphs, it is sometimes called a *uniform random graph* model Chung et al. (2001), a terminology that will also be used in this paper.

Random graphs and related structures, such maximum cliques in random graphs, have been studied intensively in last decades Erdös and Rényi (1960); Bollobás and Erdös (1976); Bollobás (2001). One of the earliest works on the asymptotical behavior of maximum clique in uniform random graphs is due to Matula Matula (1970), who showed that the maximum clique size has a strong peak around $2 \ln n / \ln(1/p)$. Grimmett and McDiarmid Grimmett and McDiarmid (1976) proved that as $n \to \infty$ the maximum clique size in a uniform random graph $G(n, p)$ is equal to $2 \ln n / \ln(1/p) + O(\ln \ln n)$ with probability one.

In Section 2, we present a generalization of this result for the case of maximum γ-cliques. Furthermore,

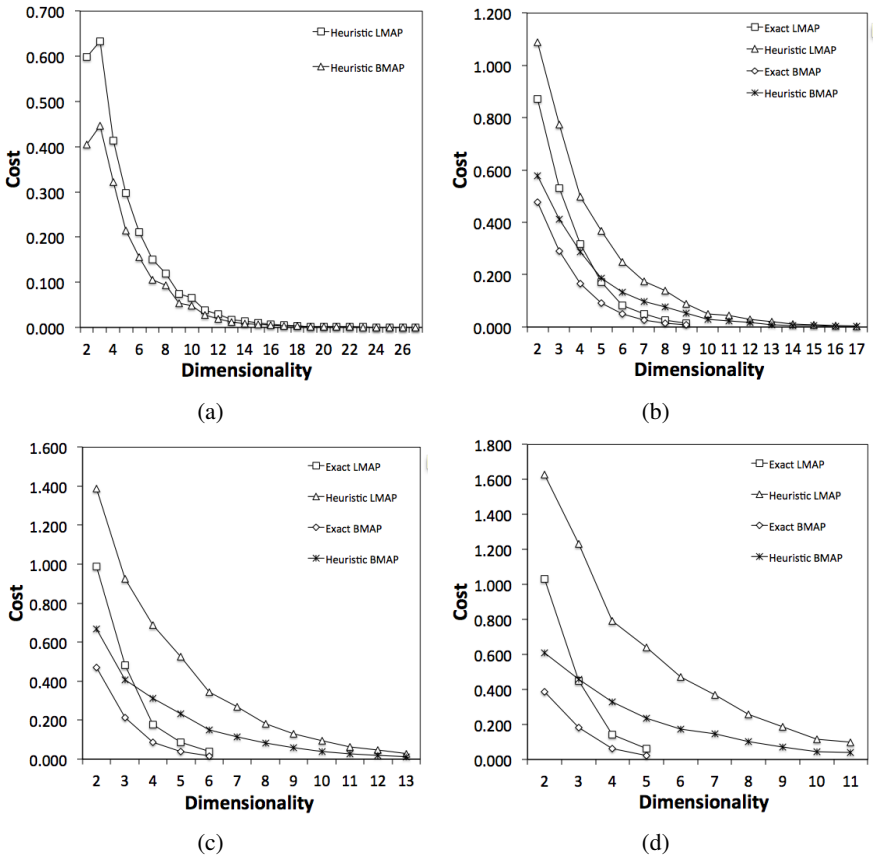


Figure 6: Comparison of the cost obtained from the heuristic method with the optimum cost in MAPs with linear sum and linear bottleneck objective functions with **(a)** $n = 2$, **(b)** $n = 3$, **(c)** $n = 4$, and **(d)** $n = 5$

we demonstrate that the size of maximum γ-clique in $G(n, p)$ undergoes a *phase transition* when the value of p is varied in the vicinity of the (fixed) value $\gamma \in (0, 1)$, manifested in a sudden and drastic change of size of the maximum γ-clique in $G(n, p)$ relative to the size of the graph itself. The phenomena of phase transition in random structures are well known in many fields of science and engineering; some of the relevant works include Bollobás et al. (2005, 2007); Łuczak (1996); Łuczak and Łuczak (2006); Łuczak (1990); Łuczak et al. (1994); Frank and Martel (1995); Krishnamachari et al. (2001).

In Section 3, we develop a linear mixed integer programming (MIP) formulation for the problem of identifying the maximum clique problem in a given graph. In comparison to traditional formulations existing in the literature for, e.g., maximum clique problem, and involving either a non-convex quadratic constraints or a number of linear constraints that is quadratic in the size of the graph, our formulation is linear and employs number of variables and constraints that is linear in the size of the graph.

5.1 Asymptotic behavior of γ-cliques in uniform random graphs

Define N_k^γ as the (random) number of γ-cliques of size k in $G(n, p)$. Noting that there are $\binom{n}{k}$ different subgraphs of size k in this graph, let I_j^γ be the indicator variable such that $I_j^\gamma = 1$ if the j-th subgraph

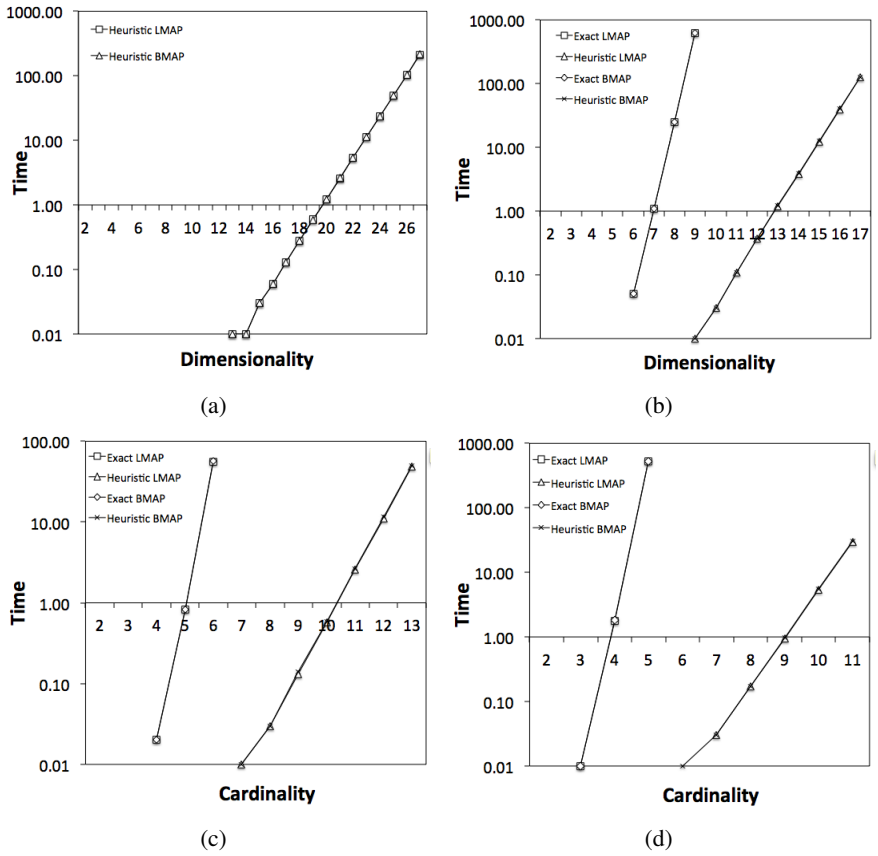


Figure 7: Comparison of the computational time in logarithmic scale needed for the optimal method and the heuristic method in MAPs with linear sum and linear bottleneck objective functions with **(a)** $n = 2$, **(b)** $n = 3$, **(c)** $n = 4$, and **(d)** $n = 5$

of size k is a γ-clique, $j = 1, \ldots, \binom{n}{k}$, and $I_j^\gamma = 0$ otherwise, which allows us to express N_n^γ as

$$N_k^\gamma = \sum_{j=1}^{\binom{n}{k}} I_j^\gamma \,. \tag{50}$$

Obviously, the unconditional probabilities $\mathsf{P}\{I_j^\gamma = 1\}$ that any subgraph of size k is a γ-clique are equal, whence the expected number of γ-cliques of size k in $G(n, p)$ is given by

$$\begin{aligned} \mathsf{E}\big[N_k^\gamma\big] &= \binom{n}{k} \mathsf{P}\{\text{a subgraph of size } k \text{ in } G \text{ is a } \gamma\text{-clique}\} \\ &= \binom{n}{k} \sum_{m=\left\lceil \gamma\binom{k}{2} \right\rceil}^{\binom{k}{2}} \binom{\binom{k}{2}}{m} p^m (1-p)^{\binom{k}{2}-m} \\ &= \binom{n}{k} \left[1 - \mathrm{Bin}\left(\left\lfloor \gamma\tbinom{k}{2} \right\rfloor; \tbinom{k}{2}, p\right)\right], \end{aligned} \tag{51}$$

where $\mathrm{Bin}(k; n, p)$ is the c.d.f. of the binomial distribution. As it will be seen, the integer $k = k_n^\gamma$ such that

$$\mathsf{E}\big[N_{k_n^\gamma}^\gamma\big] = 1 \tag{52}$$

for large values of n plays a central role in the sequel. The next proposition takes a first step in evaluating k_n^γ.

Proposition 4. *If $p < \gamma$, the integer $k = k_n^\gamma$ that satisfies $\mathsf{E}\big[N_{k_n^\gamma}^\gamma\big] = 1$ increases with n in such a way that $k_n^\gamma = o(n)$, $n \gg 1$.*

Proof. From expression (51) it is evident that $k = k_n^\gamma$ cannot be bounded for large values of n, since in that case the right-hand side of (51) would be equal asymptotically to $O(n^k)$. To verify that $k_n^\gamma = o(n)$, we construct an upper bound on the right hand side of equation (51). Using Stirling's approximation, the binomial coefficient in (51) can be bounded as

$$\binom{n}{k} \leq \Big(\frac{en}{k}\Big)^k. \tag{53}$$

To bound the summation term in (51), we use Chernoff's bound for the tail of the binomial distribution Erdös and Spencer (1974):

$$\sum_{i=m}^{n} \binom{n}{i} p^i (1-p)^{n-i} \leq \Big(\frac{n-np}{n-m}\Big)^{n-m} \Big(\frac{np}{m}\Big)^m,$$

where $m \geq np$. In our case $n = \binom{k}{2}, m = \left\lceil \gamma\binom{k}{2} \right\rceil$ (for simplicity, we use $m = \gamma\binom{k}{2}$), and since $p < \gamma$, then $m \geq np$; thus, the requirement on m is valid. Thus,

$$\sum_{i=\left\lceil \gamma\binom{k}{2} \right\rceil}^{\binom{k}{2}} \binom{\binom{k}{2}}{i} p^i (1-p)^{\binom{k}{2}-i} \leq \left(\frac{\binom{k}{2} - \binom{k}{2}p}{\binom{k}{2} - \gamma\binom{k}{2}}\right)^{\binom{k}{2}-\gamma\binom{k}{2}} \left(\frac{\binom{k}{2}p}{\gamma\binom{k}{2}}\right)^{\gamma\binom{k}{2}} =$$

$$= \left[\left(\frac{1-p}{1-\gamma}\right)^{1-\gamma} \left(\frac{p}{\gamma}\right)^{\gamma}\right]^{\binom{k}{2}}. \tag{54}$$

Combining the upper bounds in (53) and (54), we have that if $k = k_n^\gamma$ satisfies $\mathsf{E}[N_k^\gamma] = 1$, whereby the following must hold for large enough values of n:

$$1 \leq \Big(\frac{en}{k}\Big)^k \left[\left(\frac{1-p}{1-\gamma}\right)^{1-\gamma} \left(\frac{p}{\gamma}\right)^{\gamma}\right]^{\binom{k}{2}}. \tag{55}$$

Taking logarithm of the right hand side of the above inequality and dividing by k^2, we obtain

$$\frac{1}{k} + \frac{\ln n}{k} - \frac{\ln k}{k} + \frac{1}{2}\left(1 - \frac{1}{k}\right)\ln c,$$

where the constant c has the form $c = \left(\frac{1-p}{1-\gamma}\right)^{1-\gamma} \left(\frac{p}{\gamma}\right)^{\gamma}$. It is easy to see from the inequality for arithmetic and geometric means that $c \in (0,1)$ for $0 < p < \gamma < 1$. Then, if k grows with n such that $\frac{k}{\ln n} = o(1)$, the above expression becomes negative for sufficiently large n, thereby contradicting the constructed upper bound (55). This implies that $k_n^\gamma = o(n)$, which proves the proposition. ■

Proposition 5. *If $p < \gamma$, the integer $k = k_n^\gamma$ that satisfies the equality $\mathsf{E}[N_{k_n^\gamma}^\gamma] = 1$, is given by*

$$k_n^\gamma = \frac{2}{\ln\left[\left(\frac{\gamma}{p}\right)^\gamma \left(\frac{1-\gamma}{1-p}\right)^{1-\gamma}\right]} \ln n + O(\ln\ln n), \quad n \gg 1. \tag{56}$$

In establishing Proposition 5 we rely on the following result due to McKay McKay (1989).

Theorem 4 (McKay McKay (1989))**.** *Let $p \in (0,1)$ be fixed, and $p\nu \le \kappa \le \nu$ for some $\nu \ge 1$. Define $x = \frac{\kappa - p\nu}{\sigma}$, where $\sigma = \sqrt{\nu p(1-p)}$. Then*

$$\sum_{i=\kappa}^{\nu} \binom{\nu}{i} p^i (1-p)^{\nu-i} = \sigma \binom{\nu-1}{\kappa-1} p^{\kappa-1}(1-p)^{\nu-\kappa} \, \frac{1-\Phi(x)}{\phi(x)} \exp\left\{\epsilon(\nu,\kappa,p)/\sigma\right\}, \tag{57}$$

where

$$0 \le \epsilon(\nu,\kappa,p) \le \min\{\sqrt{\pi/8},\ x^{-1}\},$$

and $\Phi(x)$ and $\phi(x)$ are the cumulative and probability density functions of the standard normal distribution, respectively.

Proof of Proposition 5. Using the notations of Theorem 1, let

$$\nu = \tbinom{k}{2}, \quad \kappa = \gamma\nu, \quad \sigma = \sqrt{p(1-p)\nu}, \quad x = \frac{\gamma - p}{\sqrt{p(1-p)}} \nu^{1/2},$$

where we note that $\kappa \approx \lceil \gamma\nu \rceil > p\nu$ for large enough ν, then the last term in (57) satisfies

$$\exp\left\{\epsilon(\nu,\kappa,p)/\sigma\right\} = \exp\{O(\nu^{-1})\} = 1 + O(\nu^{-1}), \quad n \to \infty.$$

From the fact that x increases with n (cf. Proposition 4), it follows that

$$\frac{1-\Phi(x)}{\phi(x)} = \frac{1}{x} + O(x^{-3}) = \frac{\sqrt{p(1-p)}}{\gamma - p} \nu^{-1/2} \left(1 + O(\nu^{-1})\right), \quad n \to \infty,$$

where the well-known expansion

$$1 - \Phi(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2} \left(t^{-1} + O(t^{-3})\right), \quad t \to \infty,$$

was used. Invoking Stirling's expansion for $\Gamma(z)$,

$$\Gamma(z) = \sqrt{\frac{2\pi}{z}} z^z e^{-z} \big(1 + O(z^{-1})\big), \quad z \to \infty,$$

we obtain

$$\binom{\nu-1}{\gamma\nu-1} = \frac{1}{\sqrt{2\pi}} \sqrt{\frac{\gamma}{1-\gamma}} \left[\gamma^\gamma \, (1-\gamma)^{1-\gamma}\right]^{-\nu} \nu^{-1/2} \big(1 + O(\nu^{-1})\big).$$

Thus, finally, the tail of the binomial distribution in (51) can be estimated as

$$\sum_{i=\gamma\nu}^{\nu}\binom{\nu}{i}p^i(1-p)^{\nu-i}=\frac{1}{\sqrt{2\pi}}\frac{1-p}{\gamma-p}\sqrt{\frac{\gamma}{1-\gamma}}\nu^{-1/2}\left[\left(\frac{p}{\gamma}\right)^{\gamma}\left(\frac{1-p}{1-\gamma}\right)^{1-\gamma}\right]^{\nu}\left(1+O(\nu^{-1})\right),$$

where $\nu=\binom{k}{2}$. Consequently, equation (51) can asymptotically be written as

$$\binom{n}{k}\frac{1}{\sqrt{2\pi}}\frac{1-p}{\gamma-p}\sqrt{\frac{\gamma}{1-\gamma}}\binom{k}{2}^{-1/2}\left[\left(\frac{p}{\gamma}\right)^{\gamma}\left(\frac{1-p}{1-\gamma}\right)^{1-\gamma}\right]^{\binom{k}{2}}\left(1+O\left(\binom{k}{2}^{-1}\right)\right)=1. \qquad (58)$$

Taking the logarithm of both sides of the last equality, we obtain

$$-\left(k+\tfrac{1}{2}\right)\ln k+k+k\ln n+\ln c_1-\frac{1}{2}\ln\binom{k}{2}+\binom{k}{2}\ln c_2=O\left(\max\left\{\frac{k^2}{n},\frac{1}{k}\right\}\right), \qquad (59)$$

where

$$c_1=\frac{1}{2\pi}\frac{1-p}{\gamma-p}\sqrt{\frac{\gamma}{1-\gamma}},\quad c_2=\left(\frac{p}{\gamma}\right)^{\gamma}\left(\frac{1-p}{1-\gamma}\right)^{1-\gamma}.$$

Taking into account that

$$\frac{1}{2}\ln\binom{k}{2}=\ln k-\ln\sqrt{2}+O(k^{-1}),$$

equation (59) can be written as

$$\frac{k^2}{2}\ln c_2-k\ln k+k(\ln n+1-\tfrac{1}{2}\ln c_2)-\tfrac{3}{2}\ln k+\ln\sqrt{2}c_1=O\left(\max\left\{\frac{k^2}{n},\frac{1}{k}\right\}\right).$$

To obtain the main term of the asymptotical approximation of the solution of the last equation, let us restate it in the form

$$k^2\left\{\frac{1}{2}\ln c_2+\frac{\ln n}{k}-\frac{\ln k}{k}+\frac{1-\frac{1}{2}\ln c_2}{k}-\frac{3\ln k}{2k^2}+\frac{\ln\sqrt{2}c_1}{k^2}\right\}=O\left(\max\left\{\frac{k^2}{n},\frac{1}{k}\right\}\right).$$

In view of the fact that $k=o(n)$ due to Proposition 4, the above expression can be further rewritten as

$$\frac{1}{2}\ln c_2+\frac{\ln n}{k}+o\left(\frac{\ln n}{k}\right)=o(1),$$

whence we have that

$$k=\frac{2\ln n}{\ln c_2^{-1}}+\chi(n),\quad \text{where}\quad \chi(n)=o(\ln n).$$

To determine the order of the term $\chi(n)$, we restate the last equation as

$$\frac{k}{2}\ln c_2+\ln n-\ln k=O(1).$$

Writing down the expression for $k=k_n^{\gamma}$ in the form

$$k_n^{\gamma}=\frac{2\ln n}{\ln c_2^{-1}}+\chi(n),$$

and substituting it in the last equation, we obtain that $\chi(n)=O(\ln\ln n)$, which furnishes the statement of the proposition. ■

Next, we demonstrate that the number k_n^γ (56), which solves the equation $\mathsf{E}\big[N_k^\gamma\big] = 1$, with probability 1 represents an upper bound on the size of the maximum quasi-clique in a uniform random graph when $n \to \infty$. For this, we need the following property of γ-cliques.

Proposition 6. *If graph $G = (V, E)$, where $|V| = n$, is a γ-clique for some fixed $\gamma \in (0, 1]$, then for any $s < n$ there exists a γ-clique of size s in G.*

Proof. For $\gamma = 1$, this property is trivial. In the case of $\gamma \in (0, 1)$, it suffices to show that the statement of the proposition holds for $s = n - 1$. Since $G = (V, E)$ is a γ-clique, then

$$|E| \geq \gamma \binom{n}{2}.$$

Assume that there exists a vertex $i \in V$ with $\deg_G(i) \leq \gamma(n-1)$. Let $V_i = V \setminus i$; then the induced subgraph $G[V_i] = (V_i, E_i)$ is also a γ-clique, since

$$|E_i| \geq \gamma \binom{n}{2} - \gamma(n-1) = \gamma \binom{n-1}{2}.$$

If there is no such a vertex, i.e., $\deg_G(i) > \gamma(n-1)$ for all $i \in V$, then let $j = \arg\min_{i \in V} \deg_G(i)$ be the vertex of G with the smallest degree, $\deg_G(j) = m > \gamma(n-1)$. As before, denote $V_j = V \setminus j$, and observe that the cardinality of the set of edges E_j of the induced subgraph $G[V_j] = (V_j, E_j)$ satisfies

$$\begin{aligned} |E_j| &= \frac{1}{2} \sum_{i \in V_j \colon (i,j) \in E} \deg_{G[V_j]}(i) + \frac{1}{2} \sum_{i \in V_j \colon (i,j) \notin E} \deg_{G[V_j]}(i) \\ &\geq \frac{1}{2} m(m-1) + \frac{1}{2}(n-m-1)m \\ &= \frac{1}{2}(n-2)m > \gamma \binom{n-1}{2}, \end{aligned}$$

thus verifying the statement of the proposition. ■

Proposition 7. *Let M_n^γ denote the (random) size of the maximum γ-clique in a uniform random graph $G(n, p)$. If $0 < p < \gamma \leq 1$, then*

$$\limsup_{n \to \infty} \frac{M_n^\gamma}{\ln n} \leq \frac{2}{\ln 1/p^*} \qquad a.s., \tag{60}$$

where $p^* = \left(\frac{p}{\gamma}\right)^\gamma \left(\frac{1-p}{1-\gamma}\right)^{1-\gamma}$.

Proof. First, observe that

$$\mathsf{P}\{M_n^\gamma \geq k\} = \mathsf{P}\{N_k^\gamma \geq 1\} \leq \mathsf{E}[N_k^\gamma], \tag{61}$$

where the equality is due to Proposition 6. Define a sequence $k_n = \dfrac{2}{\ln 1/p^*} \ln n$, $n \geq 1$; then, from expression (51) for $\mathsf{E}[N_n^\gamma]$, one obtains by following the steps in Proposition 5 that for sufficiently large values of n

$$\mathsf{P}\{M_n^\gamma \geq k_n\} \leq \binom{n}{k_n} (p^*)^{\binom{k_n}{2}} \leq \frac{n^{k_n}}{\Gamma(k_n+1)} (p^*)^{\binom{k_n}{2}}. \tag{62}$$

From the definition of k_n it follows that the term

$$n^{k_n - 1} (p^*)^{\binom{k_n}{2}}$$

is bounded for large enough n, whence the sought probability can be subsequently bounded as

$$\mathsf{P}\{M_n^\gamma \geq k_n\} \leq \frac{n}{\Gamma(k_n+1)} \leq \frac{n e^{k_n}}{{k_n}^{k_n}}.$$

Again recalling the definition of k_n, we note that

$$n^2 \frac{n e^{k_n}}{k_n^{k_n}} = \frac{\exp\left\{k_n(1 + \frac{3}{2}\ln 1/p^*)\right\}}{k_n^{k_n}} \to 0, \quad n \to \infty, \tag{63}$$

which implies that $\mathsf{P}\{M_n^\gamma \geq k_n\} = o(n^{-2})$ for $n \gg 1$, whereby the upper bound (60) on the size of the maximum γ-clique holds almost surely by virtue of the Borel-Cantelli lemma. ■

The next corollary shows that in sufficiently large random graphs $G(n, p)$, the size of the maximum γ-clique is almost surely above a certain value of the order of $\ln n$. It uses a well known fact, established by Grimmett and McDiarmid Grimmett and McDiarmid (1976), that the size of the maximum clique in a uniform random graph $G(n, p)$ converges almost surely to $2 \ln n / \ln p^{-1}$. Note that M_n^1 represents the size of the maximum clique ($\gamma = 1$) in a uniform random graph $G(n, p)$. Observe also that, according to (56),

$$\lim_{\gamma \to 1-0} k_n^\gamma = \frac{2 \ln n}{\ln 1/p} + O(\ln \ln n),$$

which corresponds to the well-known expression for the size of the maximum clique in uniform random graphs Bollobás and Erdös (1976); Grimmett and McDiarmid (1976). This allows us to define k_n^1 as the limiting value of k_n^γ above.

Corollary 4.1. *If $0 < p < \gamma \leq 1$, then the size M_n^γ of the maximum γ-clique in a uniform random graph $G(n, p)$ satisfies*

$$\frac{2}{\ln 1/p} \leq \liminf_{n\to\infty} \frac{M_n^\gamma}{\ln n} \leq \limsup_{n\to\infty} \frac{M_n^\gamma}{\ln n} \leq \frac{2}{\ln\left[\left(\frac{\gamma}{p}\right)^\gamma \left(\frac{1-\gamma}{1-p}\right)^{1-\gamma}\right]} \qquad \text{a. s.} \tag{64}$$

where k_n^γ is given by (56).

Proof. This follows immediately from Proposition 7, and the observation that, for any $\gamma < 1$, the size of the maximum γ-clique in $G(n, p)$ always greater than the size of the maximum clique in the same graph, i.e.,

$$\mathsf{P}\{M_n^1 \leq M_n^\gamma\} = 1, \quad \gamma < 1.$$

Also note that the relations $0 < p < \gamma < 1$ imply that

$$\left(\frac{\gamma}{p}\,\frac{1-p}{1-\gamma}\right)^{1-\gamma} > 1 > \gamma,$$

from which one infers the inequality

$$\left(\frac{p}{\gamma}\right)^{\gamma}\left(\frac{1-p}{1-\gamma}\right)^{1-\gamma} > p,$$

verifying that inequality for the lower and upper bounds on $M_n^{\gamma}/\ln n$ in (64) always holds, given the above assumptions on the values of p and γ. ■

Remark 3. From Fatou's lemma it follows that bounds (64) on the maximum γ-clique size M_n^{γ}, which hold with probability 1, also hold for the average maximum γ-clique size, $\mathsf{E}[M_n^{\gamma}]$:

$$\frac{2}{\ln 1/p} \le \liminf_{n\to\infty} \frac{\mathsf{E}[M_n^{\gamma}]}{\ln n} \le \limsup_{n\to\infty} \frac{\mathsf{E}[M_n^{\gamma}]}{\ln n} \le \frac{2}{\ln\left[\left(\frac{\gamma}{p}\right)^{\gamma}\left(\frac{1-\gamma}{1-p}\right)^{1-\gamma}\right]}. \tag{65}$$

In such a way, we have established that for any fixed $\gamma > p$ the asymptotic size of the maximum γ-clique is of the order of $\ln n$. Intuitively, when $\gamma \le p$, the entire graph $G(n, p)$ becomes a γ-clique, thus the size of the maximum γ-clique has the order of n. Therefore, the natural question arising here is what happens when γ is fixed and p approaches γ. We show that there is a first order phase transition in the asymptotic behavior of the order of magnitude of the maximum γ-clique in the point $\gamma = p$.

Proposition 8. *If M_n^{γ} is the size of the maximum γ-clique in a uniform random graph $G(n, p)$ for some fixed $\gamma \in (0, 1)$, then with high probability (w.h.p.)*

$$\lim_{p\nearrow\gamma}\lim_{n\to\infty} \frac{M_n^{\gamma}}{n} = 0, \tag{66a}$$

but

$$\lim_{p\searrow\gamma}\lim_{n\to\infty} \frac{M_n^{\gamma}}{n} = 1. \tag{66b}$$

Proof. The first limiting case follows from Proposition 7, since we proved that for any fixed $\gamma > p$ with probability 1

$$\lim_{n\to\infty} \frac{M_n^{\gamma}}{n} = 0.$$

To prove the equality in (66b), let X_{ij} be a Bernoulli random variable which is equal to 1 if there exists an edge (i, j) in the uniform random graph $G(n, p)$. Then,

$$\mathsf{P}\{M_n^{\gamma} = n\} = \mathsf{P}\left\{\sum_{(i,j)} X_{ij} \ge \gamma\binom{n}{2}\right\} = \mathsf{P}\left\{\frac{\sum\limits_{(i,j)} X_{ij}}{\binom{n}{2}} \ge \gamma\right\}.$$

From the weak law of large numbers it follows that for any fixed $\varepsilon > 0$,

$$\mathsf{P}\left\{\left|\frac{\sum\limits_{(i,j)} X_{ij}}{\binom{n}{2}} - p\right| \leq \varepsilon\right\} \to 1, \quad n \to \infty\,.$$

Letting $\varepsilon = p - \gamma$, we obtain

$$\begin{aligned}\mathsf{P}\left\{\frac{\sum\limits_{(i,j)} X_{ij}}{\binom{n}{2}} \geq \gamma\right\} &= \mathsf{P}\left\{p - \frac{\sum\limits_{(i,j)} X_{ij}}{\binom{n}{2}} \leq \varepsilon\right\} \\ &\geq \mathsf{P}\left\{\left|p - \frac{\sum\limits_{(i,j)} X_{ij}}{\binom{n}{2}}\right| \leq \varepsilon\right\} \to 1, \quad n \to \infty.\end{aligned}$$

Therefore,

$$\mathsf{P}\{M_n^{\gamma} = n\} \to 1, \quad n \to \infty,$$

which ends the proof of the proposition. ■

Remark 4. The phase transition, a phenomenon of a drastic change in some property of a random structure over a small change in the structure's parameters, is well known in the literature. With respect to random graphs, the limiting probability of a graph's property changing from 0 to 1 or vice versa is well known for *monotone* and *first order* graph properties Alon and Spencer (2000). Recall that property $\mathcal{Q}$ is *monotone increasing* (respectively, *decreasing*) if from $A \subseteq B$ (resp., $B \subseteq A$) and $A \in \mathcal{Q}$ it follows that $B \in \mathcal{Q}$. The first order graph properties are ones that can be finitely described in a first order language, i.e., language consisting of variables that represent graph vertices, equality ($=$) and adjacency ($\sim$) relations, Boolean symbols $\vee$, $\wedge$, $\neg$, and the universal and existential quantifications $\forall$, $\exists$. Note that first order properties are not necessarily monotone and vice versa; for instance, the increasing property "graph is connected" cannot be expressed in first order language Janson et al. (2000). Then, limiting relations similar to (66) that concern random graphs with first order properties $\mathcal{Q}$ are known as *zero-one laws* Alon and Spencer (2000); Janson et al. (2000):

$$\lim_{n\to\infty} \mathsf{P}\{G(n, p) \text{ has } \mathcal{Q}\} = 0 \text{ or } 1,$$

where the probability is monotone if $\mathcal{Q}$ is monotone.

In this context, it is worth noting that the property that "graph is a γ-clique" in neither monotone, nor first order property, hence the phase transition in the relative size of γ-clique in uniform random graphs (66) may not be obtained directly from the general zero-one laws relations.

5.2 Linear mixed-integer formulations of the maximum γ-clique problem

In this section we develop linear mixed-integer formulations of the maximum γ-clique problem. To the best of our knowledge, the formulation that we present at the end of this section is the most compact linear formulation, of the size of $O(n)$, where n is the number of vertices in the graph.

Consider a graph $G = (V, E)$ with n vertices and an adjacency matrix A, and suppose we select some subgraph G_s of G. In order to verify whether G_s is a γ-clique, we introduce the binary vector of variables

$x \in \{0,1\}^n$, where $x_i = 1$ if vertex i belongs to G_s, and $x_i = 0$ otherwise. The subgraph G_s is a γ-clique if the cardinality of its set of edges is at least

$$\begin{aligned}\gamma\binom{|G_s|}{2} &= \frac{1}{2}\gamma\sum_{i=1}^{n} x_i\left(\sum_{i=1}^{n} x_i - 1\right) = \frac{1}{2}\gamma\left(\sum_{i,j=1}^{n} x_j x_i - \sum_{i=1}^{n} x_i\right)\\ &= \frac{1}{2}\gamma\left(\sum_{\substack{i,j=1\\ i\neq j}}^{n} x_j x_i + \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i\right)\\ &= \frac{1}{2}\gamma\sum_{\substack{i,j=1\\ i\neq j}}^{n} x_j x_i,\end{aligned}$$

where the last equality is due to $x_i^2 = x_i$. The number of edges in the subgraph G_s can be calculated as

$$\frac{1}{2}x^t Ax = \sum_{\substack{i,j=1\\ i\neq j}}^{n} a_{ij} x_j x_i.$$

Therefore, the problem of finding the maximum γ-clique in the graph G can be formulated as follows:

$$\begin{aligned}\max_{x\in\{0,1\}^n} \quad & \sum_{i=1}^{n} x_i\\ \text{s. t.} \quad & \sum_{\substack{i,j=1\\ i\neq j}}^{n} a_{ij} x_j x_i \geq \gamma \sum_{\substack{i,j=1\\ i\neq j}}^{n} x_j x_i.\end{aligned} \tag{67}$$

This is a 0–1 integer programming (IP) problem with a linear objective and a nonconvex quadratic constraint. A linearization of this problem can be performed at the expense of introducing additional variables and constraints. To this end, define $w_{ij} = x_i x_j$ for every pair of nodes (i, j). Note that only $n(n-1)/2 - n$ of such variables are required since $w_{ij} = w_{ji}$. Also, the constraint $w_{ij} = x_i x_j$ is equivalent to

$$\begin{aligned} w_{ij} &\leq x_i,\\ w_{ij} &\leq x_j,\\ w_{ij} &\geq x_i + x_j - 1.\end{aligned}$$

Now, we can reformulate (67) as a linear problem

$$\max \quad \sum_{i=1}^{n} x_i \tag{68a}$$

$$\text{s. t.} \quad \sum_{\substack{i,j=1\\ i\neq j}}^{n} a_{ij} w_{ij} \geq \gamma \sum_{\substack{i,j=1\\ i\neq j}}^{n} w_{ij}, \tag{68b}$$

$$w_{ij} \leq x_i, \qquad i, j = 1, \ldots, n, \tag{68c}$$

$$w_{ij} \leq x_j, \qquad i, j = 1, \ldots, n, \tag{68d}$$

$$w_{ij} \geq x_i + x_j - 1, \qquad i, j = 1, \ldots, n, \tag{68e}$$

$$x_i, w_{ij} \in \{0,1\}, \qquad i, j = 1, \ldots, n. \tag{68f}$$

This problem is a linear 0–1 problem with $n(n-1)/2$ variables and $\frac{3}{2}n(n-1)+1$ constraints. We can further rewrite it in a more compact form using the fact that $x_i = 0$ implies $w_{ij} = 0$, $j = i+1, .., n$. Thus, instead of (68c) and (68d) we may write

$$\sum_{j=i+1}^{n} w_{ij} \leq n x_i, \quad i = 1, \ldots, n-1,$$

which reduces the number of constraints to $\frac{1}{2}n(n-1)+n$.

In what follows, we present an improved mixed-integer linear formulation of (67) with only $O(n)$ variables and constraints. Recall that originally we had only one constraint,

$$\sum_{\substack{i,j=1 \\ i \neq j}}^{n} (a_{ij} - \gamma) x_j x_i \geq 0,$$

which we may rewrite as

$$\sum_{i=1}^{n} x_i \left(\gamma x_i + \sum_{j=1}^{n} (a_{ij} - \gamma) x_j \right) \geq 0.$$

Let us define the variables $w_i \in \mathbb{R}$, $i = 1, \ldots, n$, as follows

$$w_i = x_i \left(\gamma x_i + \sum_{j=1}^{n} (a_{ij} - \gamma) x_j \right), \quad i = 1, \ldots, n.$$

Next, observe that each of the quadratic equalities above is equivalent to four linear inequalities

$$\begin{gathered}
w_i \leq n x_i, \\
w_i \geq -n x_i, \\
w_i \geq \gamma x_i + \sum_{j=1}^{n} (a_{ij} - \gamma) x_j - (1 - x_i) n, \\
w_i \leq \gamma x_i + \sum_{j=1}^{n} (a_{ij} - \gamma) x_j + (1 - x_i) n, \\
x_i \in \{0, 1\}, \ w_i \in \mathbb{R}.
\end{gathered}$$

Therefore, the problem of finding a maximum γ-clique can be represented as the following mixed integer linear programming problem with $2n$ variables (n binary variables and n continuous variables) and $4n+1$

constraints:

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{n} x_i \\
\text{s. t.} \quad & \sum_{i=1}^{n} w_i \geq 0, \\
& w_i \leq n x_i, && i = 1, \ldots, n, \\
& w_i \geq -n x_i, && i = 1, \ldots, n, \\
& w_i \geq \gamma x_i + \sum_{j=1}^{n} (a_{ij} - \gamma) x_j - (1 - x_i) n, && i = 1, \ldots, n, \\
& w_i \leq \gamma x_i + \sum_{j=1}^{n} (a_{ij} - \gamma) x_j + (1 - x_i) n, && i = 1, \ldots, n, \\
& x_i \in \{0, 1\}, \ w_i \in \mathbb{R}, && i = 1, \ldots, n.
\end{aligned}
\tag{69}
$$

5.3 Computational experiments

In this section we illustrate the theoretical developments presented above using numerical computations of the maximum γ-clique in randomly generated graphs. In particular, the numerical studies are aimed at elucidating the following two aspects: (i) whether the asymptotic bounds (64), (65) on the size of the maximum γ-clique M_n^γ and its mean value hold for relatively small values of n, and (ii) the manifestation of the phase transition effect in the order of magnitude of the maximum γ-clique in finite-size random graphs when p approaches γ.

According to Remark 3, in large enough random graphs $G(n, p)$ the average size $\mathsf{E}[M_n^\gamma]$ of the maximum γ-clique belongs to the interval

$$
\left[m_n^1, m_n^\gamma\right] = \left[\frac{2 \ln n}{\ln\left(\frac{1}{p}\right)}, \ \frac{2 \ln n}{\ln\left(\frac{\gamma}{p}\right)^\gamma \left(\frac{1-\gamma}{1-p}\right)^{1-\gamma}} \right], \tag{70}
$$

provided that $p < \gamma$. Therefore, it was of interest to determine the applicability of the above bounds for relatively small values of n.

To this end, in the first set of computational experiments we generated a number of instances of uniform random graphs $G(n, p)$ with $n = 100$ and p ranging from 0.05 to 0.15; namely, we generated 100 instances of $G(100, p)$ for every p. Then, we employed the MIP formulation (69) to find the maximum γ-cliques in the generated graphs for values $\gamma = 0.9$ and $\gamma = 0.85$. Such a choice of parameters is justified by relatively better numerical tractability of the MIP problem (69) for sparse graphs. We used FICO$^{\text{TM}}$ Xpress Optimization Suite 7.1 Xpress to solve the resulting instances of problem (69). The resulting average values of M_{100}^γ, as well as the minimum and maximum values of M_{100}^γ over 100 instances for each p, are reported in Table 8.

In the second set of computational experiments, we analyzed the behavior of the relative size of the maximum γ-clique for a fixed γ and different values of p and n. We used $\gamma = 0.70$ and two sequences of p values: $\{p_1, \ldots, p_{100}\} = \{0.006, 0.012, \ldots, 0.600\}$ and $\{p_{101}, \ldots, p_{200}\} = \{0.601, 0.602, \ldots, 0.700\}$.

Note the increased "density" of the second sequence, which allows for a more thorough investigation of the maximum γ-clique size when the value of p becomes close to γ. For each value of $n = 500$, 1000, 5000, 1000, 20000 and p_i from the sequence defined above, we generated instances of uniform random graphs $G(n, p_i)$, and determined the maximum γ-clique size M_n^γ. Since the MIP formulation (69) becomes computationally intractable for large dense graphs, we used maximum γ-clique GRASP heuristics due to Abello et al. (2002), which were reported to perform quite well in massive graphs. Figure 8 reports the results of the described computational studies, and Figure 9 presents the results of similar studies for $\gamma = 0.5$.

Recall that it was shown in Section 2 that with probability that approaches to 1 one has

$$\lim_{n\to\infty} \frac{M_n^\gamma}{n} = \begin{cases} 0, & p < \gamma, \\ 1, & p > \gamma, \end{cases}$$

or, in other words, the relative size of the maximum γ-clique in uniform random graphs as a function of the density p of the graph with high probability represents a step function as $n \to \infty$. The results presented in Figures 8 and 9 illustrate the convergence of the relative size of maximum clique to the aforementioned limits as n increases.

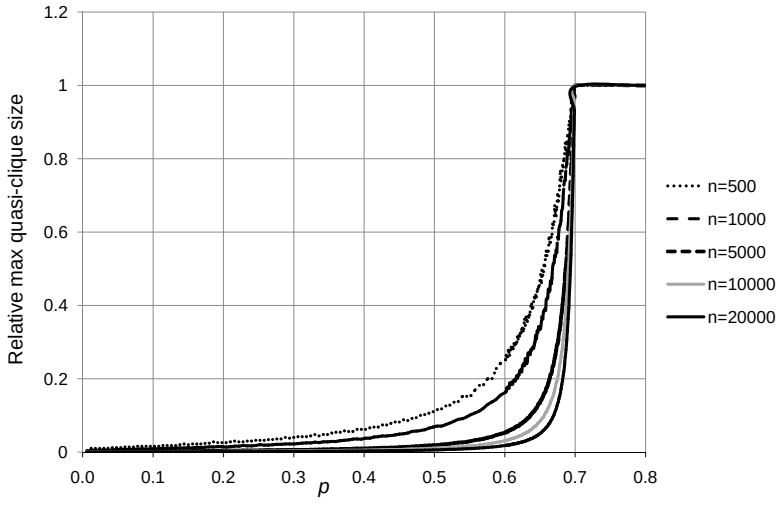


Figure 8: Relative size of the maximum γ-cliques in the uniform random graphs for $\gamma = 0.7$, $n = 500$, 1000, 5000, 1000, 20000, and $0 < p < 0.8$.

6 A bit-parallel algorithm for finding k-cliques in a k-partite graph

Given an (undirected) graph $G = (V, E)$, where V is set of nodes and E is the set of arcs, a *clique* in G is defined as a complete subset of G, i.e., a set of nodes in V that are pairwise adjacent. A clique of size k is called *k-clique*; the largest clique in a graph is called the *maximum clique* and its size is denoted by $\omega(G)$. Note that G may contain several cliques of size $\omega(G)$. Closely related to the concept of a clique is that of an *independent set* of G, defined as an induced subgraph of V whose nodes are pairwise disjoint.

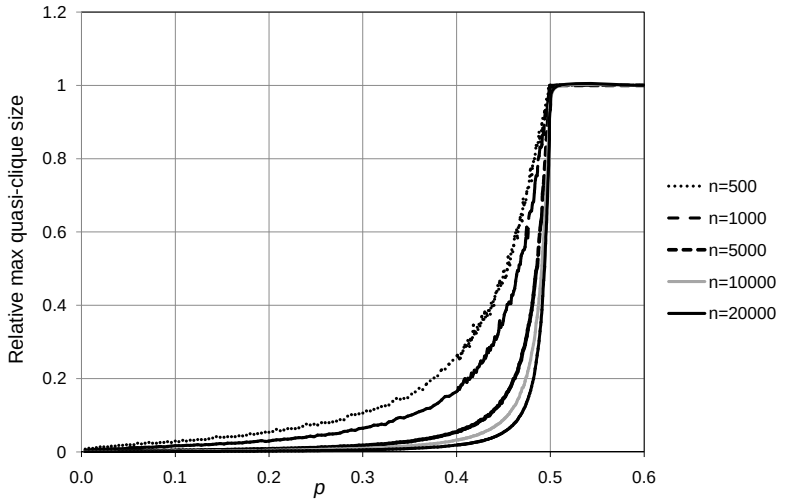


Figure 9: Relative size of the maximum γ-cliques in the uniform random graphs for $\gamma = 0.5$, $n = 500$, 1000, 5000, 1000, 20000, and $0 < p < 0.6$.

Table 8: Average ($\mathsf{E}[M_n^\gamma]$), minimum ($\underline{M}_n^\gamma$), and maximum ($\overline{M}_n^\gamma$) values of the maximum γ-clique size, computed over 100 instances of uniform random graphs $G(n, p)$ for $n = 100$ and $\gamma = 0.85$, 0.90. The lower and upper bounds m_n^1, n_n^γ are given by (70).

	$\gamma = 0.85$				$\gamma = 0.90$			
p	m_n^1	m_n^γ	$\mathsf{E}[M_n^\gamma]$	$\left[\underline{M}_n^\gamma, \overline{M}_n^\gamma\right]$	m_n^1	m_n^γ	$\mathsf{E}[M_n^\gamma]$	$\left[\underline{M}_n^\gamma, \overline{M}_n^\gamma\right]$
0.05	3.07	4.32	3.10	[3, 4]	3.07	3.88	3.10	[3, 4]
0.06	3.27	4.66	3.26	[3, 5]	3.27	4.16	3.26	[3, 5]
0.07	3.46	4.98	3.38	[3, 5]	3.46	4.44	3.38	[3, 5]
0.08	3.65	5.30	3.60	[3, 5]	3.65	4.71	3.60	[3, 5]
0.09	3.82	5.62	3.97	[3, 5]	3.82	4.97	4.97	[3, 5]
0.10	4.00	5.94	4.27	[3, 5]	4.00	5.50	4.24	[3, 5]
0.11	4.17	6.26	4.66	[4, 5]	4.17	5.77	4.65	[4, 5]
0.12	4.34	6.58	4.87	[4, 6]	4.34	6.04	4.82	[4, 5]
0.13	4.51	6.91	5.06	[5, 6]	4.51	6.04	5.00	[5, 5]
0.14	4.68	7.25	5.19	[5, 6]	4.68	6.31	5.01	[5, 6]
0.15	4.85	7.59	5.67	[5, 7]	4.85	6.59	5.06	[5, 6]

The Maximum Clique Problem (MCP) consists in finding the largest clique in a graph, and is of fundamental importance in discrete mathematics, computer science, operations research, and related fields Bomze et al. (1999). In many applications it is of interest to identify all maximum cliques in a graph. This problem is known as the Maximum Clique Enumeration Problem (MCEP). In the present work, we consider a special case of the MCEP, concerned with finding all k-cliques in a k-partite graph. A graph $G = (V, E)$ is called k-partite if the set of nodes V can be partitioned into k independent sets, or *partites* $V_r, r = 1, \ldots, k$:

$$V = \bigcup_{r=1}^{k} V_r, \quad V_r \cap V_s = \emptyset, \quad r \neq s, \text{ such that for all } i, j \in V_r : (i, j) \notin E. \tag{71}$$

Clearly, one has that $\omega(G) \leq k$ in a k-partite graph G, since the maximum clique cannot contain more than one node from each independent set V_r. The problem of finding k-cliques in k-partite graphs has found applications in textile industry Grunert et al. (2002), data mining and clustering Peters (2005), and identification of protein structures Liu and Chen (2009). This problem is not necessarily equivalent to MCEP since it does not account for maximum cliques with $\omega(G) < k$.

Grunert et al Grunert et al. (2002) proposed branch-and-bound algorithm FINDCLIQUE for the problem of finding all k-cliques in k-partite graphs, which takes as an input a graph $G = (V, E)$, where V satisfies (71), and produces the set Q of k-cliques contained in G as an output. FINDCLIQUE is a recursive method, such that level t of recursion corresponds to the level t of branch-and-bound tree, which in turn, is associated with the t-th partite that is branched on in V. Starting at the root ($t = 0$) of the branch-and-bound tree with a partial solution $S = \emptyset$, at each step of branch-and-bound procedure a node is added to or removed from S until S amounts to a k-clique in G, i.e., $|S| = k$, or it is verified that G contains no k-cliques, $\omega(G) < k$.

Let $B = \{1, \ldots, k\}$ be the index set of partites in G, $V = \bigcup_{b \in B} V_b$, and B_S denote the set of partites that have a node in S:

$$B_S = \{b \in B \mid V_b \cap S \neq \emptyset\}.$$

Given a partial solution S, a node is called *compatible* if it is adjacent to all the nodes in S; the set of compatible nodes w.r.t. S is denoted by C_S:

$$C_S = \{i \in V \mid (i, j) \in E \ \ \forall\, j \in S\}.$$

The set C_S is further partitioned into subsets containing nodes from the same partite:

$$C_S = \bigcup_{b \in \overline{B}_S} C_{S,b},$$

where $\overline{B}_S = B \setminus B_S$, and $C_{S,b} \subseteq V_b$ is given by

$$C_{S,b} = \bigcup_{s \in S} (V_b \cap N(s)),$$

with $N(s)$ being the set of nodes adjacent to node s.

At the root node of the branch-and-bound tree ($t = 0$), one has $S = \emptyset$, $B = \overline{B}_S = \{1, \ldots, k\}$, $B_S = \emptyset$, and $C_{S,b} = V_b$ for all $b \in B$. At a level t of the branch-and-bound tree, $b_t \in \overline{B}_S$ is selected as the

partition to branch on. In order to achieve the greatest reduction in the size of the branch-and-bound tree when pruning, b_t is selected as the partition with the smallest number of nodes:

$$b_t \in \arg\min_b \{|C_{S,b}| \mid b \in \overline{B}_S\}. \tag{72}$$

As long as there is a node $n_t \in C_{S,b_t}$ that is not traversed, the search process is restarted from this point with $S := S \cup \{n_t\}$ as the new partial solution. To this end, the set C_S of compatible nodes is updated with respect to $S \cup \{n_t\}$:

$$C_{S,b} := C_{S,b} \cap N(n_t) \text{ for all } b \in \overline{B}_S. \tag{73}$$

Maintaining the sets $C_{S,b}$ of nodes compatible with the current partial solution S is a key aspect of the algorithm, thus for backtracking purposes the nodes that are removed from $C_{S,b}$ during (73) are added to the set $\overline{C} = \bigcup_{t=1}^{k} \overline{C}_t$, which is similarly partitioned into k levels $\overline{C}_t$, each level corresponding to level t of the branch-and-bound tree. In other words, $\overline{C}_t$ contains the nodes in $C_{S,b}$ that are not adjacent to node n_t:

$$\overline{C}_t = \{i \in C_{S,b} \mid (i, n_t) \notin E,\ b \in \overline{B}_S\}.$$

Obviously, after this step, $C_{S,b_t} = \emptyset$. A subproblem with a partial solution S is *promising* if all of the partitions in C_S that do not share a node in the partial solution are nonempty:

$$|C_{S,b}| > 0 \text{ for all } b \in \overline{B}_S,\ b \neq b_t. \tag{74}$$

Let P be the number of partitions $C_{S,b} \subseteq C_S$ that contain at least one node; then, an upper bound on the size of the largest clique containing S is given by $|S| + P$. If $|S| + P = k$, the current subproblem is feasible, meaning S may be part of a k-clique. For a feasible subproblem, the algorithm traverses deeper into the branch-and-bound tree, $t := t + 1$, and a new subproblem is created.

Accordingly, a subproblem with partial solution S is pruned if

$$|S| + P < k, \tag{75}$$

i.e., there exists no clique of size k that contains S. For a nonpromising subproblem, set C_{S,b_t} is restored by moving the nodes in $\overline{C}_t$ back to C_S, $C_S := C_S \cup \overline{C}_t$. The last operation implicitly requires that the nodes from $\overline{C}_t$ are put back into the partitions of C_S that they were removed from:

$$C_{S,\pi(v)} := C_{S,\pi(v)} \cup v \text{ for all } v \in \overline{C}_t, \tag{76}$$

where $\pi(i)$ is the index of the partite that node i belongs to: $i \in V_{\pi(i)}$; moreover, the relative orders of nodes in the partites V_b should be preserved in $C_{S,b}$, given that the nodes in G are assumed to be ordered/numbered.

The search process is then restarted, provided that there exists a node in partition C_{S,b_t} that is not traversed. If there is no such node, FINDCLIQUE returns to the previous level $t - 1$ of the branch-and-bound tree.

6.1 A bitwise algorithm for finding k-cliques in a k-partite graph

In this section, we present an algorithm, referred to as BitCLQ, for the k-clique enumeration problem in a k-partite graph, which improves upon the FINDCLIQUE algorithm of Grunert et al Grunert et al. (2002) by introducing bitset data structures and utilizing bit parallelism for updating the set of compatible nodes and improving backtracking.

6.1.1 Bitsets

Bitsets are essentially binary vectors, or sequences of bits, and as such can be utilized efficiently in computer codes. Particularly, bitsets are useful for storing adjacency matrices of graphs, or specific subsets of ordered sets. For example, in a graph on six nodes $\{v_1, \ldots, v_6\} = V$, a clique with nodes v_1, v_2, v_3, v_5 can be represented by a bitset $\{111010\}$, where each bit corresponds uniquely to a node in the graph, with the *significant* bits (i.e., bits equal to 1) indicating the nodes in the clique. *Bit parallelism* is a form of parallel computing that achieves computational improvements by representing the problem data in bitsets of size R, where R is the machine word size (e.g., 32 or 64), such that they can be processed together within a single processor instruction. Bit parallelism has been successfully used in many computational algorithms, particularly for string matching (Grabowski and Fredriksson, 2008; Hyyrö, 2005; Hyyrö and Navarro, 2004). Recently, bit parallelism has been employed for solving hard combinatorial problems, such as SAT Segundo et al. (2008) and the Maximum Clique Problem San Segundo et al. (2011).

In the present work, bit parallelism is used to improve the computational procedure for updating the set of compatible nodes in (73), and, moreover, to achieve faster backtracking by eliminating the need for set $\overline{C}$. In addition, use of bitsets allows for improvements in memory storage efficiency for problem data structures, such as the set of compatible nodes and the adjacency matrix of the graph.

Of particular significance in the context of the present work is the operation of indexing the first significant bit in a bitset, also known as the forward bit scanning. One of the techniques for this purpose relies on use of the De Bruijn sequence with a perfect hash table Leiserson et al. (1998, http://supertech.csail.mit.edu/papers/debruijn.ps). The value to be looked up in the hash table is given by H_R below:

$$H_R := (x \wedge -x)\, D \gg (R - \log_2 R), \tag{77}$$

where x is the bitset for which the first significant bit has to be indexed, D is an instance of De Bruijn sequence, R is the machine word size, and $\gg$ stands for the binary *shift right* operator. H_R is effective for bitsets of maximum size equal to R. For larger bitsets, special containers need to be devised. The hash table required to look up the value of H_R is created based on the particular De Bruijn sequence used in (77).

Note that in (77) multiplication is performed modulo R and only the last $\log_2 R$ bits of the result will be retained. More details on forward bit scanning and the specification of the De Bruijn sequence used in (77) can be found in Leiserson et al. (1998, http://supertech.csail.mit.edu/papers/debruijn.ps).

6.1.2 BitCLQ

Below we present a modification of FINDCLIQUE, which we refer to as BitCLQ, that uses bitset data structures and bit parallelism for keeping track of the nodes in G that are compatible to the current partial solution S, while simultaneously reducing the computational cost of backtracking.

To this end, we introduce a set $\mathbb{R}$ consisting of k levels, $Z_1, \ldots, Z_k$. Each of these k levels will be used to represent the compatible nodes to the partial solution S at the t-th level of the branch-and-bound tree, where $1 \leq t \leq k$. Every level in Z is further partitioned into k sets, each corresponding to a partite V_b in G:

$$Z_t = \bigcup_{b \in B} Z_{t,b}, \quad t = 1, \ldots, k.$$

The sets $Z_{t,b}$ are represented by bitsets of size $|V_b|$. Let $Z_{t,b,i}$ be the i-th bit in $Z_{t,b}$ corresponding to

the i-th node in V_b, such that $Z_{t,b,i} = 1$ if the i-th node in V_b is compatible with all the nodes in the partial solution S at the t-th level of the branch-and-bound tree in BitCLQ:

$$Z_{t,b,i} = \begin{cases} 1, & \text{if } (i,j) \in E \text{ for all } j \in S_t; \\ 0, & \text{otherwise.} \end{cases}$$

Clearly, each level Z_t of Z is an ordered set of combination of bitsets with the total size $|V|$. Further, the adjacency matrix M of graph G is stored in the bitset form, with the convention that the i-th row (column) corresponds to the i-th bit in Z_t, $t = 1, \ldots, k$.

BitCLQ is initialized by setting $t := 0$, $S := \emptyset$, $B = \overline{B}_S := \{1, \ldots, k\}$, and $Q := \emptyset$, where Q is the set of all k-cliques in G. Note that since at the beginning all the nodes in G can be added to S to extend its size, all the bits in Z_1 are significant:

$$Z_{1,b,i} = 1 \text{ for all } b \in \overline{B}(S_t),\ i \in V_b.$$

At level t of the branch-and-bound tree, the partition b_t to branch on is selected as

$$b_t \in \arg\min_b \{|Z_{t,b}| \mid b \in \overline{B}_S\}, \tag{78}$$

where $|Z_{t,b}|$ is defined as the number of significant bits in the bitset $Z_{t,b}$. The forward bit scanning method discussed in Section 6.1.1 is used to identify node $n_t \in V_{b_t}$ that has not been traversed and thus can be added to the partial solution. As long as such a node exists in V_{b_t}, the search process is restarted with $S := S \cup \{n_t\}$ as the partial solution, and the corresponding bit in Z_{t,b_t} is set to 0.

Utilizing bitsets also facilitates the process of updating the compatible nodes: when n_t is added to partial solution, Z_{t+1} is created by performing a logical AND operation with Z_t and the row $M(n_t)$ of the adjacency matrix corresponding to the node n_t as operands:

$$Z_{t+1} = Z_t \wedge M(n_t). \tag{79}$$

Similarly to FINDCLIQUE, let P denote the number of partitions $Z_{t,b}$ with $|Z_{t,b}| > 0$ at level the t of the branch-and-bound tree. If $|S| + P = k$, the current partial solution is promising, so that a new subproblem is created, and BitCLQ proceeds one level deeper into the branch-and-bound tree, $t := t + 1$. If the partial solution is not promising, the method presented in Section 6.1.1 is used to select nodes in V_{b_t} that have not been traversed. If such a node is found, the search process is restarted, otherwise backtracking is performed by simply updating $t := t - 1$. Note that due to the special structure of Z, BitCLQ does not need to restore the set of compatible nodes during backtracking, in contrast to the update procedure (76) for the set C_S that is performed in FINDCLIQUE.

6.1.3 Example

As an illustration, consider the 3-partite graph that is shown along with its adjacency matrix M in Figure 10, where the partite 1 consists of nodes $\{1, 2, 3\}$, partite 2 contains nodes $\{4, 5, 6\}$, and partite 3 contains nodes $\{7, 8, 9\}$. BitCLQ is initialized by setting $S := \emptyset$, $\overline{B}_S := \{1, 2, 3\}$ and $Z_1 := \{111|111|111\}$. Since all the partites are of the same size, i.e. $|Z_{1,b}| = 3$ for all $b \in \overline{B}_S$, the one to branch on is chosen arbitrarily; assume that the first partite $Z_{1,1}$ is chosen for branching. The search process from this point restarts 3 times, each time adding one of the three nodes in $Z_{1,1}$. The first node to add to S is node 1, $Z_{1,1,1}$ is then set to 0, and Z_2 is subsequently created by performing logical AND operation with Z_1 and the corresponding row of the adjacency matrix M as operands:

Algorithm 2 BitCLQ(t)

1: $b_t \in \arg\min_b \{|Z_{t,b}| \mid b \in \overline{B}_S\}$
2: $i :=$ the first significant bit in Z_{t,b_t}
3: **repeat**
4: $\quad n_t :=$ the i-th node in b_t
5: $\quad Z_{t,b,i} := 0$
6: $\quad S := S \cup \{n_t\}$
7: $\quad$ **if** $|S| = k$ **then**
8: $\quad\quad Q := Q \cup S$
9: $\quad\quad S := S \setminus \{n_t\}$
10: $\quad$ **else**
11: $\quad\quad Z_{t+1,b} := Z_{t,b} \wedge M(n_t)$ for all $b \in \overline{B}_S$
12: $\quad\quad B_S := B_S \cup \{b_t\};\ \overline{B}_S := \overline{B}_S \setminus \{b_t\}$
13: $\quad\quad P :=$ number of partitions $Z_{t,b}$ with $|Z_{t,b}| > 0, b \in \overline{B}_S$
14: $\quad\quad$ **if** $|S| + P = k$ **then**
15: $\quad\quad\quad$ BitCLQ($t+1$)
16: $\quad\quad\quad S := S \setminus \{n_t\}$
17: $\quad\quad\quad B_S := B_S \setminus \{b_t\};\ \overline{B}_S := \overline{B}_S \cup \{b_t\}$
18: $\quad\quad$ **else**
19: $\quad\quad\quad S := S \setminus \{n_t\}$
20: $\quad\quad\quad B_S := B_S \setminus \{b_t\};\ \overline{B}_S := \overline{B}_S \cup \{b_t\}$
21: $\quad\quad$ **end if**
22: $\quad$ **end if**
23: $\quad i :=$ the first significant bit in Z_{t,b_t}
24: **until** $i \leq |V_{b^t}|$

$t := 1$,
$S := \{1\}$,
$Z_2 := Z_1 \wedge M(1) = \{011|111|111\} \wedge \{000|111|011\} = \{000|111|011\}$,
$\overline{B}_S := \{2, 3\}$.

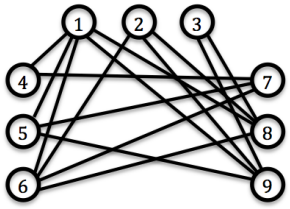
As a result, the set Z_2 of nodes compatible with the partial solution $S = \{1\}$ contains nodes $\{4, 5, 6, 8, 9\}$. Since none of the partites in $\overline{B}_S$ is empty, the partial solution S is promising and a new subproblem is created. The objective in the new subproblem is to find a $|\overline{B}_S|$-clique in Z_2. A node from $Z_{2,3}$ will be added to S (since $|Z_{2,3}| < |Z_{2,2}|$). The first node in $Z_{2,3}$ to add to the partial solution is node 8. The bit corresponding to node 8 is set $Z_{2,3,2} := 0$, and we have

$t := 2$,
$S := \{1, 8\}$,
$Z_3 := Z_2 \wedge M(8) = \{000|111|001\} \wedge \{111|001|000\} = \{000|001|000\}$,
$\overline{B}_S := \{2\}$.

Again, the partites in $\overline{B}_S$ contain at least 1 node (node 6) in Z_3. So the partial solution is promising, and a new subproblem is created. In the next step, node 5 is added to S:

$t := 3$,
$S := \{1, 8, 6\}$.

At this point, since $|S| = k = 3$, i.e., a k-clique is found. To continue the search for other k-cliques, the last node in S is removed. BitCLQ searches $Z_{3,2}$ for another node that can be added to S. Since such a



$$M = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} \begin{array}{c} \begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Figure 10: A 3-partite graph and its adjacency matrix.

node does not exist, the algorithm backtracks: $t := 2$, node 8 is removed from S, and BitCLQ restarts with $S = \{1, 9\}$ as the partial solution.

6.2 Numerical Results

In order to illustrate the performance of the proposed method, the k-clique enumeration problem for k-partite graphs has been solved by BitCLQ and FINDCLIQUE for randomly generated graph instances of several types. Both algorithms were implemented in C++ and ran on a 64-bit Windows machine with 3GHz dual-core processor and 4GB of RAM. It is worth noting that the original implementation of FINDCLIQUE algorithm by Grunert et al Grunert et al. (2002) relies on the use of `vectors` and `links` data types from the C++ standard template library (STL). In our experiments, we observed that by replacing the original data structure of vectors of lists with arrays, up to 300% improvement in FINDCLIQUE running time is achieved on the data sets used in our case study. The numerical results reported for the FINDCLIQUE algorithm are obtained using this "improved" implementation.

Our numerical experiments involve randomly generated instances of k-partite graphs of two types. The first set of instances consists of two groups: small-size instances and large-size instances. In the small-size instances, k-partite graphs are randomly generated with the number of partites in the range $k \in [3, 10]$. For each value of k, the reported running times and the number of k-cliques in the graph are averaged over 10 instances. Table 9 shows the summary of the experimental results for this first group. The columns of the table show the number k of partites in the k-partite graph, the number m of nodes in each partite of the graph, the total number $|V|$ of nodes in the graph, the graph's density p, and the total number of k-cliques in the graph (#CLQ). The density parameter p is used for generation of the graphs, and is equal to the probability of an edge connecting two nodes from different partites: $\Pr\{(v_i, v_j) \in E\} = p$.

The second group include instances of larger size with the values of $k \in \{25, 50, 75, 100\}$. For each value of k in this group, 10 random instances of the k-partite graph have been generated and solved by FINDCLIQUE and BitCLQ. Table 10 summarizes the results of the experiments for this group. Since the graphs used in this set of experiments are rather large and the list of all k-cliques contained in them may not be found in a reasonable time, the solution process has been terminated after 200 seconds and the number of k-cliques found by each method was recorded. BitCLQ outperformed FINDCLIQUE in all cases.

The second set of experiments was conducted to compare the performance of BitCLQ with FIND-CLIQUE on randomly generated instances of Multidimensional Assignment Problem (MAP). As shown in Krokhmal et al. (2007); Krokhmal and Pardalos (2011); Mirghorbani et al., high-quality solutions for

Table 9: Average computational time (in seconds) to find all the k-cliques (#CLQ) contained in randomly generated k-partite graphs.

k	m	$\|V\|$	p	#CLQ	FINDCLIQUE	BitCLQ
3	100	300	0.1	1004	0.005	0.002
4	100	400	0.15	1124	0.008	0.002
5	100	500	0.2	1047	0.015	0.003
6	100	600	0.25	939	0.031	0.006
7	50	350	0.35	192	0.009	0.004
8	50	400	0.4	299	0.021	0.007
9	50	450	0.45	683	0.055	0.021
10	50	500	0.5	2672	0.176	0.071

Table 10: Average number of k-cliques found in randomly generated instances of k-partite graphs after 200 seconds.

k	m	$\|V\|$	p	time	FINDCLIQUE	BitCLQ
25	40	1000	0.8	200	13,556,733	23,516,581
50	30	1500	0.9	200	800,369	1,032,111
75	30	2250	0.95	200	557,042,389	735,722,241
100	30	3000	0.95	200	348,416	365,799

randomized MAPs can be obtained as n-cliques in n-partite graphs that are constructed in a special way from the problem's data (in this case, n denotes the number of elements per dimension in a d-dimensional MAP). For MAPs with random iid costs, the resulting n-partite graph can be viewed as randomly generated with a certain density. The corresponding results are reported in Table 11, where n denotes the number of partitions in the graphs, and d is the number of dimensions d in the MAP.

References

Abello, J., Resende, M. G. C., and Sudarsky, S. (2002) "Massive Quasi-Clique Detection," in: "LATIN 2002: Theoretical Informatics," 598–612, Lecture Notes in Computer Science, Springer.

Alba, R. D. (1973) "A graph-theoretic definition of a sociometric clique," *Journal of Mathematical Sociology*, **3**, 113–126.

Albrecher, H. (2005) "A note on the asymptotic behaviour of bottleneck problems," *Operations Research Letters*, **33** (2), 183–186.

Albrecher, H., Burkard, R. E., and Çela, E. (2006) "An asymptotical study of combinatorial optimization problems by means of statistical mechanics," *Journal of Computational and Applied Mathematics*, **186** (1), 148–162.

Aldous, D. (1992) "Asymptotics in the random assignment problem," *Probability Theory and Related Fields*, **93** (4), 507–534.

Table 11: Average computational time (in seconds) needed to find the first n-clique in an n-partite graph corresponding to a randomized instance of the Multidimensional Assignment Problem with d dimensions and n elements per dimension.

n	d	m	$\|V\|$	p	BitCLQ	FINDCLIQUE
10	3	10	100	0.74	0.00	0.00
20	3	12	240	0.86	0.00	0.00
30	3	13	390	0.91	0.02	0.00
40	3	13	520	0.93	0.76	1.38
50	3	14	700	0.94	0.42	0.42
60	3	14	840	0.95	55.28	86.87
70	3	14	980	0.96	251.78	395.34
10	4	22	220	0.65	0.00	0.00
20	4	28	480	0.82	0.08	0.20
30	4	31	930	0.87	8.18	22.41
10	5	48	480	0.59	0.00	0.01
20	5	68	1360	0.77	13.29	28.23

Aldous, D. (2001) "The $\zeta(2)$ limit in the random assignment problem," *Random Structures and Algorithms*, **18** (4), 381–418.

Alon, N. and Spencer, J. (2000) *The Probabilistic Method*, Wiley, New York, 2nd edition.

Andrijich, S. M. and Caccetta, L. (2001) "Solving the multisensor data association problem," *Nonlinear Analysis*, **47** (8), 5525–5536.

Anstreicher, K. M. (2003) "Recent advances in the solution of quadratic assignment problems," *Mathematical Programming*, **97** (1–2), 27–42.

Balas, E. and Saltzman, M. J. (1991) "An algorithm for the three-index assignment problem," *Operations Research*, **39** (1), 150–161.

Balasundaram, B., Butenko, S., and Hicks, I. V. (2011) "Clique Relaxations in Social Network Analysis: The Maximum k-plex Problem," *Operations Research*, to appear.

Berge, C. (1989) *Hypergraphs: combinatorics of finite sets*, North-Holland, Amsterdam.

Bollobás, B. (1998) *Modern graph theory*, Springer, New York.

Bollobás, B. (2001) *Random Graphs*, Cambridge University Press.

Bollobás, B. and Erdös, P. (1976) "Cliques in random graphs," *Mathematical Proceedings of Cambridge Philosophical Society*, **80**, 419–427.

Bollobás, B., Janson, S., and Riordan, O. (2005) "The phase transition in the uniformly grown random graph has infinite order," **26**, 1–36.

Bollobás, B., Janson, S., and Riordan, O. (2007) "The phase transition in inhomogeneous random graphs," *Random Structures and Algorithms*, **31** (1), 3–122.

Bomze, I. M., Budinich, M., Pardalos, P. M., and Pelillo, M. (1999) "The Maximum Clique Problem," in: "Handbook of Combinatorial Optimization," 1–74, Kluwer Academic Publishers.

Buck, M. W., Chan, C. S., and Robbins, D. P. (2002) "On the Expected Value of the Minimum Assignment," *Random Structures & Algorithms*, **21** (1), 33–58.

Burkard, R. E. (2002) "Selected topics on assignment problems," *Discrete Applied Mathematics*, **123** (1–3), 257–302.

Burkard, R. E. and Çela, E. (1999a) "Linear assignment problems and extensions," in: D.-Z. Du and P. M. Pardalos (Eds.) "Handbook of Combinatorial Optimization, Supplement Volume A," 75–149, Kluwer Academic Publishers, Dordrecht.

Burkard, R. E. and Çela, E. (1999b) "Quadratic and three-dimensional assignments," in: M. Dell'Amico, F. MQoli, and S. Martello (Eds.) "Annotated Bibliographies in Combinatorial Optimization," 373–391, Wiley, Chichester.

Burkard, R. E., Çela, E., and Klinz, B. (1994) "On the Biquadratic Assignment Problem," in: P. M. Pardalos and H. Wolkowicz (Eds.) "Quadratic Assignment and Related Problems," volume 16 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 117–146, American Mathematical Society, Providence, RI.

Burkard, R. E., Dell'Amico, M., and Martello, S. (2009) *Assignment Problems*, SIAM, Philadelphia, PA.

Burkard, R. E. and Fincke, U. (1982a) "The Asymptotic Probabilistic Behavior of Quadratic Sum Assignment Problems," *Zeitschrift für Operations Research*, **27**, 73–81.

Burkard, R. E. and Fincke, U. (1982b) "On random quadratic bottleneck assignment problems," *Mathematical Programming*, **23** (2), 227–232.

Burkard, R. E. and Fincke, U. (1985) "Probabilistic asymptotic properties of some combinatorial optimization problems," *Discrete Applied Mathematics*, **12** (1), 21–29.

Chummun, M. R., Kirubarajan, T., Pattipati, K. R., and Bar-Shalom, Y. (2001) "Fast data association using multidimensional assignment with clustering," *IEEE Transactions on Aerospace and Electronic Systems*, **37** (3), 898–913.

Chung, F., Aiello, W., and Lu, L. (2001) "A random graph model for power law graphs," *Experimental Mathematics*, **10**, 53–66.

Coppersmith, D. and Sorkin, G. (1999) "Constructive bounds and exact expectations for the random assignment problem," *Random Structures and Algorithms*, **15** (2), 113–144.

Crama, Y. and Spieksma, F. C. R. (1992) "Approximation algorithms for three-dimensional assignment problems with triangle inequalities," *European Journal of Operational Research*, **60** (3), 273–279.

Donath, W. E. (1969) "Algorithm and average-value bounds for assignment problems," *IBM Journal of Research and Development*, 380–386.

Dutta, A. and Tsiotras, P. (2008) "Egalitarian peer-to-peer satellite refueling strategy," *Journal of Spacecraft and Rockets*, **45** (3), 608–618.

Dyer, M. E., Frieze, A. M., and McDiarmid, C. J. H. (1986) "On linear programs with random costs," *Mathematical Programming*, **35** (1), 3–16.

Erdös, P. (1947) "Some remarks on the theory of graphs," *Bulletin of American Mathematical Society*, **53**, 292–294.

Erdös, P. and Rényi, A. (1960) "On the Evolution of Random Graphs," *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, **5**, 17–61.

Erdös, P. and Spencer, J. H. (1974) *Probabilistic Methods in Combinatorics*, Academic Press, New York.

Feller, W. (1968) *An Introduction to Probability Theory and Its Applications*, volume I, John Wiley & Sons, New York, 3rd edition.

Frank, J. and Martel, C. U. (1995) "Phase Transitions in the Properties of Random Graphs," in: "Principles and Practice of Constraint Programming (CP-95," 62–69.

Frenk, J. B. G., van Houweninge, M., and Rinnooy Kan, A. H. G. (1985) "Asymptotic properties of the quadratic assignment problem," *Mathematics of Operations Research*, **10** (1), 100–116.

Frenk, J. B. G., van Houweninge, M., and Rinnooy Kan, A. H. G. (1987) "Order Statistics and the Linear Assignment Problem," *Computing*, **39** (2), 165–174.

Goemans, M. X. and Kodialam, M. S. (1993) "A lower bound on the expected value of an optimal assignment," *Mathematics of Operations Research*, **18** (2), 267–274.

Grabowski, S. and Fredriksson, K. (2008) "Bit-parallel string matching under Hamming distance in $O(n\lceil m/w\rceil)O(n\lceil m/w\rceil)$ worst case time," *Information Processing Letters*, **105** (5), 182–187.

Grimmett, G. R. and McDiarmid, C. J. H. (1976) "On colouring random graphs," *Mathematical Proceedings of Cambridge Philosophical Society*, **77**, 313–324.

Grundel, D., Krokhmal, P., Oliveira, C., and Pardalos, P. (2007) "On the number of local minima in the multidimensional assignment problem," *Journal of Combinatorial Optimization*, **13** (1), 1–18.

Grunert, T., Irnich, S., Zimmermann, H., Schneider, M., and Wulfhorst, B. (2002) "Finding all k-cliques in k-partite graphs, an application in textile engineering," *Computers & Operations Research*, **29** (1), 13–31.

Hahn, P. M., Kim, B.-J., Stuetzle, T., Kanthak, S., Hightower, W. L., Samra, H., Ding, Z., and Guignard, M. (2008) "The quadratic three-dimensional assignment problem: Exact and approximate solution methods," *European Journal of Operational Research*, **184** (2), 416–428.

Hamming, R. W. (1950) "Error Detecting and Error Correcting Codes," *Bell System Technical Journal*, **26** (2), 147–160.

Hyyrö, H. (2005) "Bit-parallel approximate string matching algorithms with transposition," *Journal of Discrete Algorithms*, **3** (2–4), 215–229.

Hyyrö, H. and Navarro, G. (2004) "Bit-Parallel Witnesses and Their Applications to Approximate String Matching," *Algorithmica*, **41** (3), 203–231.

Janson, S., Łuczak, T., and Rucinski, A. (2000) *Random Graphs*, John Wiley & Sons, Inc., New York.

Jonker, R. and Volgenant, A. (1987) "A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problems," *Computing*, **38**, 325–340.

Karp, R. M. (1972) "Reducibility among combinatorial problems," in: R. E. Miller and J. W. Thatcher (Eds.) "Complexity of Computer Computations," 85–103, Plenum Press, New York.

Karp, R. M. (1987) "An upper bound on the expected cost of an optimal assignment," in: D. S. Johnson, T. Nishizeki, A. Nozaki, and H. S. Wilf (Eds.) "Discrete Algorithms and Complexity," volume 15 of *Perspectives in Computing*, 1–4, Academic Press, Boston.

Kirubarajan, T., Wang, H., Bar-Shalom, Y., and Pattipati, K. R. (2001) "Efficient multisensor fusion using multidimensional data association," *IEEE Transactions on Aerospace and Electronic Systems*, **37** (2), 386–400.

Kravtsov, V. M. (2005) "Polynomial algorithms for finding the asymptotically optimum plan of the multiindex axial assignment problem," *Cybernetics and Systems Analysis*, **41** (6), 940–944.

Krishnamachari, B., Wicker, S. B., and Béjar, R. (2001) "Phase Transition Phenomena in Wireless Ad-Hoc Networks," in: "IN GLOBECOM," .

Krokhmal, P., Grundel, D., and Pardalos, P. (2007) "Asymptotic Behavior of the Expected Optimal Value of the Multidimensional Assignment Problem," *Mathematical Programming*, **109** (2–3), 525–551.

Krokhmal, P. and Pardalos, P. M. (2009) "Random Assignment Problems," *European Journal of Operational Research*, **194** (1), 1–17.

Krokhmal, P. A. and Pardalos, P. M. (2011) "Limiting optimal values and convergence rates in some combinatorial optimization problems on hypergraph matchings," *Submitted for publication*.

Lazarus, A. J. (1993) "Certain expected values in the random assignment problem," *Operations Research Letters*, **14** (4), 207–214.

Leiserson, C. E., Prokop, H., and Randall, K. H. (1998, http://supertech.csail.mit.edu/papers/debruijn.ps) "Using de Bruijn Sequences to Index a 1 in a Computer Word," *Working paper*.

Linusson, S. and Wästlund, J. (2000) "A generalization of the random assignment problem," ArXiv:math.CO/0006146.

Linusson, S. and Wästlund, J. (2004) "A proof of Parisi's conjecture on the random assignment problem," *Probability Theory and Related Fields*, **128** (3), 419–440.

Liu, Q. and Chen, Y. (2009) "High functional coherence in k-partite protein cliques of protein interaction networks," *Bioinformatics and Biomedicine*.

Loiola, E. A., Maia de Abreu, N. M., Boaventura-Netto, P. O., Hahn, P. M., and Querido, T. (2007) "A survey for the quadratic assignment problem," *European Journal of Operational Research*, **176** (2), 657–690.

Łuczak, M. J. and Łuczak, T. (2006) "The phase transition in the cluster-scaled model of a random graph," *Random Structures and Algorithms*, **28**, 215–246.

Łuczak, T. (1990) "Component Behavior Near the Critical Point of the Random Graph Process," *Random Structures and Algorithms*, **1**, 287–310.

Łuczak, T. (1996) "The Phase Transition in a Random Graph," in: D. Miklós, V. T. Sós, and T. Szönyi (Eds.) "Combinatorics, Paul Erdös is Eighty," volume 2, 399–422, Bolyai Mathematical Society, Budapest.

Łuczak, T., Pittel, B., and Wierman, J. C. (1994) "The Structure of a Random Graph at the Point of the Phase Transition," *Transactions of the American Mathematical Society*, **341** (2), 721–748.

Martin, O. C., Mézard, M., and Rivoire, O. (2005) "Random multi-index matching problems," *Journal of Statistical Mechanics*, P09006.

Matsumoto, R., Kurosawa, K., Itoh, T., Konno, T., and Uyematsu, T. (2006) "Primal-dual distance bounds of linear codes with application to cryptography," *IEEE Transactions on Information Theory*, **52** (9), 4271–4256.

Matula, D. W. (1970) "On the complete subgraphs of a random graph," in: "Proceedings of the 2nd Conference on Combinatorial Mathematics and its Applications," 356–369, University of North Carolina, Chapel Hill.

McKay, B. D. (1989) "On Littlewood's Estimate for the Binomial Distribution," *Advances in Applied Probability*, **21** (2), 475–478.

Merz, P. and Freisleben, B. (2000) "Fitness landscape analysis and memetic algorithms for the quadratic assignment problem," *IEEE Transactions on Evolutionary Computation*, **4** (4), 337–352.

Mézard, M. and Parisi, G. (1985) "Replicas and optimization," *Journal de Physique Lettres*, **46** (17), 771–778.

Mirghorbani, M., Krokhmal, P., and Pasiliao, E. L. (????) "Computational Studies of Randomized Multidimensional Assignment Problems," in: A. Sorokin, M. T. Thai, and P. M. Pardalos (Eds.) "Dynamics of Information Systems," in press, Springer.

Mokken, R. J. (1979) "Cliques, clubs and clans," *Quality and Quantity*, **13** (2), 161–173.

Murphey, R., Pardalos, P., and Pitsoulis, L. (1998a) "A greedy randomized adaptive search procedure for the multitarget multisensor tracking problem," in: P. M. Pardalos and D.-Z. Du (Eds.) "Network Design: Connectivity and Facilities Location," volume 40 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 277–302, American Mathematical Society.

Murphey, R., Pardalos, P., and Pitsoulis, L. (1998b) "A Parallel GRASP for the Data Association Multidimensional Assignment Problem," in: "Parallel Processing of Discrete Problems," volume 106 of *The IMA Volumes in Mathematics and its Applications*, 159–180, Springer, Berlin.

Nair, C., Prabhakar, B., and Sharma, M. (2005) "Proofs of the Parisi and Coppersmith-Sorkin random assignment conjectures," *Random Structures & Algorithms*, **27** (4), 413–444.

Olin, B. (1992) *Asymptotic properties of the random assignment problem*, PhD thesis, Department of Mathematics, Royal Institute of Technology, Stockholm, Sweden.

Papadimitrou, C. H. and Steiglitz, K. (1998) *Combinatorial Optimization: Algorithms and Complexity*, Dover, New York.

Pardalos, P. M. and Pitsoulis, L. (Eds.) (2000) *Nonlinear Assignment Problems: Algorithms and Applications*, Kluwer Academic Publishers, Dordrecht.

Pardalos, P. M. and Ramakrishnan, K. G. (1993) "On the expected optimal value of random assignment problems: Experimental results and open questions," *Computational Optimization and Applications*, **2** (3), 261–271.

Parisi, G. (1998) "A conjecture on random bipartite matching," *arXiv:cond-mat/9801176v1*.

Pentico, D. W. (2007) "Assignment problems: A golden anniversary survey," *European Journal of Operational Research*, **176** (2), 774–793.

Peters, M. (2005) "CLICK: Clustering categorical data using k-partite maximal cliques," *IEEE International Conference on Data Engineering*.

Pferschy, U. (1996) "The random linear bottleneck assignment problem," *RAIRO – Operations Research*, **30** (2), 127–142.

Pierskalla, W. (1968) "The multidimensional assignment problem," *Operations Research*, **16** (2), 422–431.

Poore, A. B. (1994a) "Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking," *Computation Optimization and Applications*, **3** (1), 27–54.

Poore, A. B. (1994b) "A numerical study of some data association problems arising in multitarget tracking," in: W. W. Hager, D. W. Hearn, and P. M. Pardalos (Eds.) "Large Scale Optimization: State of the Art," 339–361, Kluwer Academic Publishers, Dordrecht.

Poore, A. B. and Gadaleta, S. (2006) "Some assignment problems arising from multiple target tracking," *Mathematical and Computer Modelling*, **43** (9–10), 1074–1091.

Poore, A. B. and Robertson, A. J. (1997) "A new Lagrangian relaxation based algorithm for a class of multidimensional assignment problems," *Computation Optimization and Applications*, **8** (2), 129–150.

Pusztaszeri, J. F., Rensing, P. E., and Liebling, T. M. (1996) "Tracking elementary particles near their primary vertex: a combinatorial approach," *Journal of Global Optimization*, **9** (1), 41–64.

Rhee, W. T. (1988) "A note on asymptotic properties of the quadratic assignment problem," *Operations Research Letters*, **7** (4), 197–200.

Rhee, W. T. (1991) "Stochastic analysis of the quadratic assignment problem," *Mathematics of Operations Research*, **16** (2), 223–239.

Sahni, T. and Gonzales, T. (1976) "P-Complete Approximation Problems," *Journal of the Association for Computing Machinery*, **23** (3), 555–565.

Samra, H., Ding, Z., and Hahn, P. M. (2005) "Symbol mapping diversity design for multiple packet transmissions," *IEEE Transactions on Communications*, **53** (5), 810–817.

San Segundo, P., Rodríguez-Losada, D., and Jiminez, A. (2011) "An exact bit-parallel algorithm for the maximum clique problem," *Computers & Operations Research*, **38** (2), 571–581.

Scott, J. (2007) *Social Network Analysis: A Handbook*, SAGE Publications, London, 2nd edition.

Segundo, P. S., Tapia, C., Puente, J., and Rodrguez-Losada, D. (2008) "A New Exact Bit-Parallel Algorithm for SAT." in: "ICTAI (2)'08," 59–65.

Spieksma, F. C. R. (2000) "Multi index assignment problems: complexity, approximation, applications," in: L. Pitsoulis and P. M. Pardalos (Eds.) "Nonlinear Assignment Problems: Algorithms and Applications," 1–12, Kluwer Academic Publishers, Dordrecht.

Stadler, P. F. (1996) "Landscapes and their correlation functions," *Journal of Mathematical Chemistry*, **20** (1), 1–45.

Stanley, R. (1986) *Enumerative Combinatorics*, Wadsworth & Brooks, Belmont, CA.

Szpankowski, W. (1995) "Combinatorial optimization problems for which almost every algorithm is asymptotically optimal!" *Optimization*, **33** (4), 359–367.

Veenman, C. J., Hendriks, E. A., and Reinders, M. J. T. (1998) "A fast and robust point tracking algorithm," in: "Proceedings of the Fifth IEEE International Conference on Image Processing," 653–657, Chicago.

Walkup, D. W. (1979) "On the expected value of a random assignment problem," *SIAM Journal on Computation*, **8** (3), 440–442.

Wasserman, S. and Faust, K. (1994) *Social Network Analysis*, Cambridge University Press.

Wästlund, J. (2005a) "A proof of a conjecture of Buck, Chan, and Robbins on the expected value of the minimum assignment," *Random Structures and Algorithms*, **26** (1–2), 237–251.

Wästlund, J. (2005b) "A simple proof of the Parisi and Coppersmith-Sorkin formulas for the random assignment problem," *Linköping Studies in Mathematics*, (6).

Weinberger, E. D. (1990) "Correlated and uncorrelated fitness landscapes and how to tell the difference," *Biological Cybernetics*, **63** (5), 325–336.

Xpress (2010) "FICO™ Xpress Optimization Suite 7.1," `http://www.fico.com/en/Products/DMTools/Pages/FICO-Xpress-Optimization-Suite.aspx`.