Maximum Cut-Clique Problem: ILS Heuristics and a Data Analysis Application

Pedro Martins[(1,3)*], Antonio Ladrón[(2)], Helena Ramalhinho[(2,3)]

[(1)] ISCAC, Polytechnic Institute of Coimbra, Coimbra, Portugal
[(2)] Department of Economics and Business, Business Analytics Research Group, Universitat Pompeu Fabra, Barcelona, Spain
[(3)] Operations Research Center (CIO), Faculty of Sciences, University of Lisbon, Portugal

Abstract

This paper focuses on iterated local search heuristics for the maximum cut-clique (or clique neighborhood) problem. Given an undirected graph $G=(V,E)$ and a clique C of G, the cut-clique is the set of edges running between C and $V \backslash C$, establishing the cut $(C, V \backslash C)$. The maximum cut-clique in G is to find a clique with the largest number of edges in the neighborhood of the clique, also known as the maximum edge-neighborhood clique. This problem has been recently introduced in the literature together with a number of applications, namely in cell biology instances. However, it has only been addressed so far by exact methods.

In this paper, we introduce the first approximate algorithms for tackling the maximum cut-clique problem, compare the results with the exact methodologies and explore a new application within marketing analysis, providing a new alternative perspective for mining market basket problems.

Keywords: Cut-cliques, clique's edge neighborhood, iterated local search heuristics, discretized formulations, market basket analysis, data mining.

1. Introduction

Searching for dense components in a network has long been attracting many researchers from different areas. Among those structures, there is the concept of a clique, in which all

* Corresponding author. Address: ISCAC – Quinta Agrícola – Bencanta, 3040-316 Coimbra, Portugal. Tel.: +351 239 802 000; fax: +351 239 445 445.
E-mail addresses: pmartins@iscac.pt (P. Martins), helena.ramalhinho@upf.edu (H. Ramalhinho), antonio.ladron@upf.edu (A. Ladrón).

elements are pairwise adjacent. This structure is expected to reveal a strongly related set of elements.

A large number of applications involving cliques have been discussed in the literature since long. Some of those applications can be found in coding theory, fault diagnosis, computer vision, pattern recognition (Bomze, Budinich, Pardalos & Pelillo, 1999), telecommunications, marketing, e-commerce (Cavique, 2007; Raeder & Chawla, 2011), financial markets, social networks and in molecular and biological networks (Bull, Muldoon & Doig, 2013; Strickland, Barnes & Sokol, 2005; Spirin & Mirny, 2003).

To formalize the problem, let $G=(V,E)$ be an undirected graph, where $V=\{1,\ldots,n\}$ is the set of nodes and $E\subseteq V\times V$ the set of edges. A clique of G is a subset of nodes $C\subseteq V$ whose elements are pairwise adjacent, that is, $(i,j)\in E$ for all pairs $i,j\in C$. Finding the maximum cardinality clique in G is known as the Maximum Clique (MC) problem. The cardinality of the maximum clique is the clique number of the graph, being denoted by $\omega(G)$. A maximal clique is a clique that is not a proper subset of any other clique.

A very extensive survey addressing the MC problem, up to 1999, can be found in Bomze, et al. (1999). For more recent results we can find contributions on bounding methods for the clique number of G (Gendron, Hertz & St-Louis, 2008; Luz & Schrijver, 2005) on exact enumerative algorithms (Östergard, 2002; Tomita & Kameda, 2007), on heuristics (Grosso, Locatelli & Pullan, 2008; Solnon & Fenet, 2006; Pullan & Hoos, 2006), and on formulations discussions (Martins, 2010).

The MC problem belongs to the NP-hard class (Karp, 1972). Furthermore, there is no polynomial-time approximation algorithm for it, unless P=NP, (Crescenzi, Fiorini & Silvestri, 1991). In fact, the problem is not approximable within $n^{1/4-\varepsilon}$, for any $\varepsilon > 0$, (Bellare, Goldreich & Sudan, 1998).

In the present paper we consider a different clique's related problem. Instead of searching for the largest size clique in the graph, we want a clique (of any size) with the largest number of edges incident to the nodes in the clique, excluding those within the clique. This problem has been recently introduced in Martins (2012), where formulations were proposed and showed their applicability to some real world problems. In formal terms, given a clique C of a graph $G=(V,E)$, the *edge neighborhood* (or *cut-clique*) of C is the set of all edges in the cutset $(C,V\backslash C)$, that is, $E'(C)=\{(i,j)\in E : i\in C \text{ and } j\in V\backslash C\}$. When C is a singleton, namely when $C=\{i\}$, we denote $E'(C)$ by $E'(i)$. Similarly, we denote by $N(i)$ the set of nodes adjacent to node i in G, that is, $N(i)=\{j\in V : (i,j)\in E\}$. Note that the edges that link the nodes in $N(i)$ with i are exactly those in $E'(i)$, thus, $|E'(i)|=|N(i)|$. We also denote by Γ the set of all cliques in G. Searching for a clique $C\in\Gamma$ with largest $E'(C)$ is known as the *maximum edge neighborhood clique* (MENC) *problem*, or *maximum cut-clique* (MCC) *problem*. In what follows, we use the second designation (MCC) when mentioning this problem, and use indistinctly the two terms: edge neighborhood clique and cut-clique, meaning the same. Figure 1 shows the maximum clique and the maximum cut-clique in a 14 nodes graph. The maximum clique solution includes 4 nodes (C^1 = {1, 2, 3, 4}), while the maximum cut-clique has only 3 nodes (C^2 = {6, 7, 9}). On the other hand, the total number of edges in the neighborhood of C^1 (cutset $E'(C^1)$ = 2) is much smaller than the edge neighborhood of C^2 (cutset $E'(C^2)$ = 9). In effect, the smaller sized clique (C^2) is much more engaged in the network than the largest size clique (C^1), which may suggest that the smaller sized clique can be more interactive within the whole network. Actually, in some cases, the maximum clique solution can lead us to an isolated component of the graph, being displaced from the "crowdie" zone.

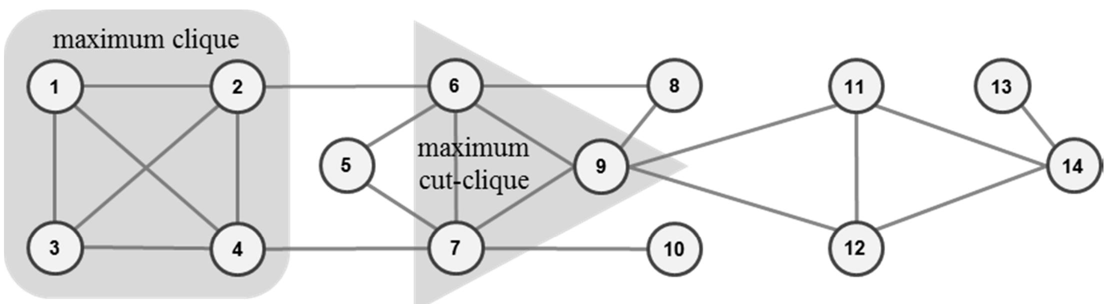


Figure 1: A maximum clique and a maximum cut-clique in a 14 nodes graph.

We describe known exact formulations from the literature and propose Iterated Local Search (**ILS**) methods based on the heuristic proposed in Grosso, et al. (2008), including specific properties of the MCC problem to accelerate the search. These heuristics were shown to be very fast and accurate when addressing the maximum clique problem, obtaining outstanding results for known benchmark instances. Another relevant feature of the ILS heuristics is their low dependency on parameterization, which is an important practical concern when handling real-world applications.

One motivation to focus on the Maximum Cut-Clique Problem comes from the application to Market Basket Analysis (MBA). In marketing, the field of Market Basket Analysis consists of identifying meaningful associations in a customer transaction dataset. The area is becoming increasingly relevant due to the amount of data that the stores and supermarkets have available today, for example from the loyalty card. The main objective of this field is to analyze large datasets of store transactions and obtain relevant insights to do a better planning of the marketing strategies and operations. The information obtained from the analysis of this data can have an important impact in the business strategy and operations, for example product placement, optimal product-line offering, personalized marketing campaigns and product promotions. Some of the common methodologies and techniques in MBA are: Association Rules, Detecting Communities, Association Rules Networks, Hyperclique Pattern Discovery and Center Piece sub-graphs. For a survey on Market Basket Analysis see (Hipp, Günther & Nakhaeizadeh, 2000; Raeder & Chawla, 2011; Zaki, 1999; Aguinis, 2013). The traditional and oldest interdependence approach for analyzing market basket data is the detection and estimation of conditioned purchase probabilities for pairs of products purchased in the same basket, known as association rules. However, consumer datasets frequently contain hundreds of association rules, so filtering and selecting the relevant subsets of interdependence patterns is not a trivial or easy task (Klemettinen, et al., 1994). In the last few years researchers have developed several techniques to address this important limitation of the traditional approach (see Raeder & Chawla (2011) for a detailed description). Among these methodologies, the analysis of network-based rules, as the cut-clique approach, can find relevant and meaningful relationships across sets of products in large consumer purchase datasets. Some examples of network-based methodologies for MBA can be found on the following works. Videla-Cavieres & Ríos (2014) present a novel approach based on graph mining techniques to the MBA and applied it to a large example of a wholesale supermarket chain. Kim et al. (2012) propose a product network analysis for MBA using a bipartite graph, meanwhile Raeder & Chawla (2009) use a more general network. Keshavamurthy et al. (2013) presents an association rule mining Genetic Algorithm and an application to MBA. Kamakura (2012) compares and contrasts traditional MBA with a sequential extension and exemplifies it with a real application.

In a different approach, Cavique (2007) proposes the search for large products-set patterns within market basket analysis using maximum weighted cliques. To the best of our knowledge, this the most similar work to the one discussed in the present paper.

In this case, each edge $(i,j)\in E$ has an associated weight w_{ij} , representing the number of times that products i and j were bought together, during the entire time range, and the cost of a clique is the sum of the weights of all its edges. Thus, the *maximum edge-weight clique* (MEWC) problem looks for a clique with maximum total cost in G, which should reveal the most frequent set of products commonly sharing (in pairs) the same basket. In effect, the MEWC can be seen as the edge-weighted version of the MC problem; and the same way, we can define the *maximum edge-weight neighborhood clique* (MEWNC) problem, in which we look for a clique with maximum edge-weight in its neighboring edges, representing the weighted version of the MCC problem. Formulations for both MEWC and MEWNC problems on sparse graphs are discussed in Gouveia & Martins (2013).

If we consider again the example in Figure 1, and set $w_{11,12} = 100$ and $w_{ij} \leq 10$ for all the remaining edges ($E \backslash \{(11,12)\}$), then the maximum edge-weight clique in the graph is $C^3 = \{11, 12, 14\}$ with cost at most 120; while the maximum edge-weight neighborhood clique is $C^4 = \{9, 11\}$ with cost at most 150. In effect, compared to the MCC optimum solution $C^2 = \{6, 7, 9\}$, the two cliques C^3 and C^4 have only 2 and 5 links in their neighborhoods, respectively, being less related to the remaining nodes in the graph than C^2. As a result, we may expect a stronger interaction between C^2 and the remaining nodes in G than from C^3 or C^4.

Considering this motivation, we use an inter-relationship network among the products and conduct our search sustaining that the linkage from one set of items (those in the clique) to all others is more relevant for finding dependencies among the products than the number of times they are selected. This is the reason for concentrating the discussion on unweighted graphs instead of doing it on their weighted counterparts, in order to avoid that strongly weighted edges may capture the search, and moving it into heavily purchased products, while missing the intended items dependencies. These dependencies can potentially concentrate interesting consumer habits, to be explored in future marketing campaigns.

The main contribution of this work is an efficient ILS based heuristic for the Cut-Clique Problem and a new alternative technique for mining information on Market Basket networks. Raeder & Chawla (2011) say "... no techniques currently available in the literature sufficiently addresses the problem of finding meaningful relationships in a large transaction databases.". We propose the use of cut-cliques models and an ILS metaheuristic that also contributes to a new and innovative approach to obtain insights and relevant association rules from a market network. The techniques proposed here can be complementary to the ones based on data mining, since both provide relevant insights about the market basket. One of the main advantages is that the proposed ILS metaheuristic can solve very large scale datasets, and so we can apply it to individual products, not only to brands of products. Finally we present an application of the cut-clique approach to a database from the British ice cream market containing information over a 2½-year period (January 2006 to June 2008) among 142 households to obtain relevant information on the household purchase behavior.

In the next section we discuss the ILS based heuristics, while exact methods for the MCC problem are described in section 3 using known formulations. Computational tests on benchmark instances are performed in section 4, and the application of these methodologies to a real-world market network is discussed in section 5. The paper ends with a section for conclusions.

2. Iterated local search algorithms

In this section we describe iterated local search (ILS) based heuristics for the maximum cut-clique problem, following the ILS algorithms proposed in Grosso, et al. (2008) for the maximum clique version. These algorithms are derived from the Dynamic Local Search methods described in Pullan & Hoos (2006). Considering the computational results reported in Grosso, et al. (2008), we detach the following reasons to sustain our choice: the accuracy of the algorithms, their speed, and low dependency on parameterization. For a survey on ILS algorithms and applications see (Lourenço, Martin & Stützle, 2003; Lourenço, Martin & Stützle, 2010).

The ILS algorithm comprises two main operations: add/aspiration moves and swap moves. The add/aspiration moves correspond to an incremental constructive process. When failing, the algorithm tries to modify this solution to a neighboring clique of equal size, if existing, by appropriately switching two nodes, one from the clique and the other one coming from its complementary set. The entering node is selected among a set of candidates that are linked to all nodes in the clique except one, called *one-missing* nodes. This process is the so called *plateau search* procedure (also considered in Battiti & Protasi (2001)), credited as one of the key elements of the algorithm's success (Grosso, et al., 2008). Restart procedures are applied when add or swap moves are no longer possible to be carried out. The restarting solution is a perturbed version of the best clique returned by the previous stage. This is a different technique from random multistart, where a completely random independent solution is generated for starting a new stage.

For describing the ILS we consider three disjunctive subsets of V:

- C is the set of nodes in the clique under construction;
- $K_0(C)$ is the set of candidate nodes for increasing the size of the clique in C, that is, $i \in K_0(C)$ if $(i,j) \in E$ for all $j \in C$;
- $K_1(C)$ is the set of one-missing nodes to C, that is, $i \in K_1(C)$ if $(i,j) \in E$ for all $j \in C \setminus \{r\}$ and $(i,r) \notin E$.

We also consider the set U of tabu nodes. These are the dropped nodes from C, as a consequence of the swap moves.

Compared with the MC problem, there is an important observation that should be stressed before handling the MCC version.

Observation: The MC problem verifies the inclusionwise property, that is, any subset S of a clique $C \in \Gamma$ (with $|C|>1$) is still a clique and $|S| \leq |C|$. However, when considering the cut-clique function to optimize in the MCC problem, the inclusion-wise property does not hold. In fact, it is not hard to find an example to which, for a given clique $C \in \Gamma$ (with $|C|>1$) and a subset $S \subseteq C$, the cut-set of S is larger than the cut-set of C, that is, $|E'(S)|>|E'(C)|$, indicating that a clique may contain a subset with larger edge neighborhood.

This observation suggests that orienting the search for a systematic increase of the size of the current clique may not lead to the right direction. Thus, the constructive scheme in the original ILS algorithm, based on a sequence of add moves, should be oriented by the cardinality of the cut-set of the putative cliques $(C \cup \{i\})$, for all $i \in K_0(C)$. Otherwise, the incremental process may deteriorate the cut-set cardinality value.

Concerning the previous observation, it is important to compare the growth of $|E'(C)|$ after adding node i to C, that is, comparing $|E'(C)|$ with $|E'(C \cup \{i\})|$, for $i \in K_0(C)$. In effect, after including node i in C, its edge neighborhood loses $|C|$ edges. In addition, the number of edges brought by the neighborhood of i to the newly set $C \cup \{i\}$ are no more than $|E'(i)|-|C|$, because $|C|$ of those edges will be kept in the interior of $C \cup \{i\}$. Thus,

$$|E'(C \cup \{i\})| = |E'(C)| - |C| + |E'(i)| - |C| = |E'(C)| + |E'(i)| - 2|C|$$

So, compared with $|E'(C)|$, the inclusion of node i into C is profitable only if $|E'(C \cup \{i\})| > |E'(C)|$, that is, only if $|E'(i)| > 2|C|$.

In fact, if the inclusion of node i into a $|C|$-sized clique is unprofitable, then its inclusion still remains unprofitable in any clique of larger cardinality. Hence, we say that C is a *maximal cut-clique* if there is no other clique in G, containing C, with larger cut-clique cardinality. These results prove the following proposition.

Proposition 1. Given a clique C of G and a node $i \in V \backslash C$. If $|E'(i)| < 2|C|$, then node i will not belong to any optimum solution to the MCC problem involving a clique with cardinality $|C|+1$ or larger. Furthermore, if $|E'(i)| = 2|C|$, the inclusion of node i will not improve clique's C cut-set cardinality.

As a result of Proposition 1, for a given clique C, any node $i \in V \backslash C$ with $|E'(i)| \leq 2|C|$ will not increase the clique's C cut-set ($|E'(C)|$), thus node i should not belong to $K_0(C)$.

Following similar arguments, we can also establish an analogous condition for one-missing nodes to a given clique C. Thus, if $i \in K_1(C)$ and j is the associated missing node in C, that is, $\{j\} = C \backslash N(i)$, then

$$|E'(C \backslash \{j\} \cup \{i\})| = |E'(C)| - |E'(j)| + 2(|C|-1) + |E'(i)| - 2(|C|-1) = |E'(C)| - |E'(j)| + |E'(i)|$$

So, this time, the inclusion of node i into C is profitable only if $|E'(C \backslash \{j\} \cup \{i\})| > |E'(C)|$, that is, only if $|E'(i)| > |E'(j)|$.

Using the previous observations, for a given clique $C \in \Gamma$, the set of candidate nodes $K_0(C)$ and one-missing nodes $K_1(C)$ are defined as

$$K_0(C) = \{ i \in V \backslash C : |C \cap N(i)| = |C| \text{ and } |E'(i)| > 2|C| \} \tag{1}$$

$$K_1(C) = \{ i \in V \backslash C : |C \cap N(i)| = |C| - 1 \text{ and } |E'(i)| > |E'(j)|, \text{ for } \{j\} = C \backslash N(i) \} \tag{2}$$

Following the description in (Grosso, et al., 2008) and considering the new characterizations for sets $K_0(C)$ and $K_1(C)$ established in (1) and (2), respectively, we present the basic ILS algorithm for the MCC problem (adapted from Algorithm 1 in Grosso, et al. (2008)). The input data are graph G and an integer value for setting parameter *max_sel*. This is the only parameter needed in the algorithm, and it controls the maximum number of modifications of set C (local/plateau search iterations), which influences the running time of the algorithm:

ILS Algorithm for the MCC problem

1. Set $C^* \leftarrow \varnothing$, $C \leftarrow \varnothing$, $sel \leftarrow 0$;
2. **while** ($sel < max\_sel$) **do**
3. Randomly select a node $i \in V \backslash C$; /* Perturbation/restart */
4. Set $C \leftarrow [C \cap N(i)] \cup \{i\}$;
5. Set $U \leftarrow \varnothing$ and $C' \leftarrow C$; /* Local/plateau search */ ↓
6. **while** ($K_0(C) \neq \varnothing$ **or** $K_1(C) \backslash U \neq \varnothing$) **and** ($C \cap C' \neq \varnothing$) **and** ($sel < max\_sel$) **do**

7. **if** $(K_0(C)\setminus U \neq \emptyset)$ **then** /* add move */
8. Select a node $i \in K_0(C)\setminus U$;
9. Set $C \leftarrow C \cup \{i\}$;
10. **if** $(U = \emptyset)$ **then** $C' \leftarrow C$;
11. **else-if** $(K_1(C)\setminus U \neq \emptyset)$ **then** /* swap move */
12. Select a node $i \in K_1(C)\setminus U$;
13. Set $C \leftarrow [C \cup \{i\}]\setminus \{j\}$, $U \leftarrow U \cup \{j\}$, where $\{j\} = C\setminus N(i)$;
14. **else-if** $(K_0(C) \cap U \neq \emptyset)$ **then** /* aspiration */
15. Select a node $i \in K_0(C) \cap U$;
16. Set $C \leftarrow C \cup \{i\}$;
17. **end-if**;
18. sel $\leftarrow$ sel + 1;
19. **end-while**;
20. **if** $(|E'(C)| > |E'(C^*)|)$ **then**
21. Set $C^* \leftarrow C$;
22. **end-if**;
23. **end-while**;
24. **return** C^*.

When $C=\emptyset$ we set $E'(C)=\emptyset$ and $|E'(C)|=0$. A complete execution of the inner while cycle (lines 6-19), up to exiting the while loop, is called a stage. This corresponds to a complete local search phase. A stage is ended when $K_0(C)=\emptyset$ and $K_1(C)\setminus U=\emptyset$, that is, when C is a maximal cut-clique and there are no more candidates for swap moves (excluding the nodes in U); or when $C \cap C'=\emptyset$, that is, when the current clique has no node in common with the first maximal cut-clique found at this stage; or when the maximum number of selections (*max_sel*) is attained. Set C' represents the first maximal cut-clique found during a stage. It is updated when the current clique C is perturbed (lines 4-5), or during the initial growing process at the beginning of a stage (line 10), before any swap move, leading to the first maximal cut-clique in the current stage. Thus, condition $(C \cap C' \neq \emptyset)$ acts as an anchor to C', guaranteeing that the search will be kept local. Otherwise, when $C \cap C'=\emptyset$, the solution (C) is forced to be perturbed (in line 4), because it is totally apart from C'.

During a stage, the algorithm performs a sequence of add and swap moves, giving priority to the constructive process (add moves), until it reaches a maximal cut-clique. Then, a swap move is performed, changing the current clique to an equal size neighboring clique, performing a plateau search procedure. From that point, the constructive process is tried again. All nodes that were removed from C during a stage are kept tabu in set U. These nodes are forbidden to re-enter C during the whole stage, unless they are shown to be profitable in a last-chance add move (those in $K_0(C)\cap U$), considered only after the failure of both add or swap moves. This last-chance device can be seen as an aspiration procedure. At the end of a stage, the current clique C is perturbed by inserting a randomly generated node. This apparently minimal perturbation can cause a significant damage in C, depending on the outcome of $C\cap N(i)$. Nevertheless, the new forthcoming set (C) can preserve most of the structure of its ancestor, which is an important difference from other techniques, namely the shaking process in Variable Neighborhood Search algorithms. Contrarily to the algorithm described in Grosso, et al. (2008), we opted to randomly select a node $i\in V\backslash C$ instead of selecting the node from the whole set V, in line 3 of the algorithm. This option is to avoid forcing the forthcoming stage to repeat the same set C.

When the selection of a node for the add, swap or aspiration moves is made randomly, then we have the totally random version of the ILS algorithm, and we denote it by **R-ILS**, corresponding to Algorithm 1 in Grosso, et al. (2008). This version is not based on node evaluation but on completely random selection.

Alternatively, Grosso, et al. (2008) also proposed the selection of nodes for leaving sets $K_0(C)$ and $K_1(C)$ based on a maximum node degree criterion (ties broken randomly). The node degree here considered is defined on the entire graph G, that is, defined by function $|N(i)|$, and not the node "residual degree" $|K_0(C)\cap N(i)|$ for $i\in K_0(C)$ (or, $|K_1(C)\cap N(i)|$ for $i\in K_1(C)$) as regarded in the notable Reactive Local Search (RLS) algorithm (Battiti & Protasi, 2001). An important advantage for using graph's G nodes degrees is that they can be computed and sorted before running the algorithm. This way, the computational effort of the algorithm is not damaged when compared with the version that involves random choices all over, because the computational effort for selecting a node is basically the same in both cases. We denote by **D-ILS** the ILS algorithm that resorts to the mentioned degree nodes sorting for selecting nodes in the add, swap or aspiration moves, performed in lines 8, 12, and 15, respectively. The computational experiments conducted in Grosso, et al. (2008) report, however, that the totally random version (R-ILS) performs better than the version that includes the mentioned deterministic selection rules (D-ILS), when addressing the MC problem.

Grosso, et al. (2008) also propose a version that incorporates penalties on the nodes of G, as a device for promoting diversification (their Algorithm 2). They also discuss alternative restart rules, involving different strategies for perturbing set C in each restarting stage.

More detailed features and computational performance concerning the ILS algorithm for the MC problem are depicted in Grosso, et al. (2008).

3. Exact methods for the maximum cut-clique problem

In this section we present exact methods based on mathematical formulation for the Maximum Cut-Clique Problem. We also propose a stronger variable elimination test and a sequential algorithm for assisting a discretized model.

3.1 Known formulations

Characterizing the set of all cliques in a graph $G=(V,E)$ is an important step for modeling clique's related problems. Due to its combinatorial nature, this set has been characterized within integer programming, using the following decision variables,

$$x_i = \begin{cases} 1 & \text{if node } i \text{ is in the clique} \\ 0 & \text{otherwise} \end{cases}, \quad \forall i \in V$$

This is the simplest variables space for modeling cliques, leading to the following natural formulation (see, (Bomze, et al., 1999))

$$x_i + x_j \leq 1 \ , \quad \forall (i,j) \in \overline{E} \qquad (3)$$

$$x_i \in \{0,1\} \ , \quad \forall i \in V \qquad (4)$$

where $\overline{E}$ represents the complementary set of edges in G, that is, $\overline{E} = \{(i,j) \in V \times V : (i,j) \notin E \wedge i \neq j\}$, and $\overline{G} = (V, \overline{E})$ represents the complementary graph.

A more compact formulation was proposed in Della Croce & Tadei (1994). Instead of constraints (3), the authors considered the following set of inequalities

$$\sum_{j \in \overline{N}(i)} x_j \leq \left|\overline{N}(i)\right|(1 - x_i) \ , \quad \forall i \in V \qquad (5)$$

with $\overline{N}(i)$ representing the set of nodes adjacent to node i in $\overline{G}$, that is, $\overline{N}(i) = \left\{ j \in V : (i,j) \in \overline{E} \right\}$. Inequalities (5) can be shown to be an aggregated version of constraints (3). In spite of being more compact, this representation can be weaker than the former, from a linear programming (LP) relaxation standpoint.

More recently, an extended and discretized formulation for characterizing cliques was proposed in Martins (2010). It uses the following sets of decision variables

$$w^q = \begin{cases} 1 & \text{if the clique size is equal to } q \\ 0 & \text{otherwise} \end{cases}, \quad \forall q \in Q$$

$$v_i^q = \begin{cases} 1 & \text{if the clique size is equal to } q \text{ and node } i \text{ belongs to the clique} \\ 0 & \text{otherwise} \end{cases}, \quad \forall i \in V, \ \forall q \in Q$$

with $Q=\{q_{min},\ldots,q_{max}\}$ a set of consecutive clique's sizes. When addressing the MC problem, this range includes $\omega(G)$, that is, $1 \le q_{min} \le \omega(G) \le q_{max}$. The mentioned formulation can be described by

$$\sum_{j \in N(i)} v_j^q \ge (q-1) v_i^q \ , \quad \forall i \in V \ , \ \forall q \in Q \tag{6}$$

$$\sum_{i \in V} v_i^q = q w^q \ , \quad \forall q \in Q \tag{7}$$

$$\sum_{q \in Q} w^q = 1 \tag{8}$$

$$v_i^q \in \{0,1\} \ , \quad \forall i \in V \ , \ \forall q \in Q \tag{9}$$

$$w^q \in \{0,1\} \ , \quad \forall q \in Q \tag{10}$$

Considering the computational experiments conducted in Martins (2010) for the MC problem, the three characterizations should be used according to the density of G. Hence, (3) should be the appropriate model for addressing very dense graphs, and (6)-(8) should be the right choice for very sparse graphs. The in-between instances should be handled by model (5).

Formulations for the MCC problem were discussed in Martins (2012). Most of those formulations use the previously mentioned clique's characterizations. This problem involves finding a clique in G with maximum edge neighborhood (or cutset), that is, $\max_{C \in \Gamma} \left\{ \left(\sum_{i \in C} |E'(i)| \right) - |C| \cdot \left(|C| - 1 \right) \right\}$, which forces the models to know the size of the clique ($|C|$). This can be accomplished using the set of variables $\{w^q \ (q \in Q)\}$, leading to the following three formulations for the MCC problem

FMCC1: $$\max_{(x,w)} \left\{ \sum_{i \in V} |E'(i)| x_i - \sum_{q \in Q} q \cdot (q-1) w^q \ : \ (3), (7'), (8) \ \text{and} \ (x,w) \in \{0,1\}^{n+|Q|} \right\}$$

FMCC2: $$\max_{(x,w)} \left\{ \sum_{i \in V} |E'(i)| x_i - \sum_{q \in Q} q \cdot (q-1) w^q \ : \ (5), (7'), (8) \ \text{and} \ (x,w) \in \{0,1\}^{n+|Q|} \right\}$$

Equality (7') is defined by $\sum_{i \in V} x_i = \sum_{q \in Q} q w^q$, being the $\{x\}$ variables version of equalities (7).

FMCC3: $$\max_{(v,w)}\left\{\sum_{i\in V}\sum_{q\in Q}\left|E'(i)\right|v_i^q-\sum_{q\in Q}q\cdot(q-1)w^q\ :\ (6),(7),(8)\ \text{and}\ (v,w)\in\{0,1\}^{(n+1)|Q|}\right\}$$

with $x=\{x_1,\ldots,x_n\}$, $w=\{w^{q_{\min}},\ldots,w^{q_{\max}}\}$ and $v=\{v_1^{q_{\min}},\ldots,v_1^{q_{\max}},v_2^{q_{\min}},\ldots,v_2^{q_{\max}},\ldots,v_n^{q_{\min}},\ldots,v_n^{q_{\max}}\}$. Remember that the set of edges incident to i and the set of nodes adjacent to i have the same cardinality, $|E'(i)|=|N(i)|$.

Considering again the computational experiments reported in Martins (2010, 2012), the three models should be considered according to the density of the graph, using model FMCC1 for very dense graphs and model FMCC3 for very sparse graphs.

3.1 Variable elimination test

In order to reduce the size of the models, we propose a new variable elimination test, arising from the following corollary taken from Proposition 1.

Corollary 2. Given a node $i\in V$ and a clique size $q\in Q$, if $|E'(i)|\leq 2(q-1)$ (or $|N(i)|\leq 2(q-1)$), then variable v_i^q can be removed from model FMCC3.

The new test is denoted by Test 2, while a former test proposed in Martins (2012), based on condition $|N(i)|<(q-1)$ is denoted by Test 1. As $2(|C|-1)>|C|-1$ for any $C\in\Gamma$ with $|C|>1$, then it is easy to show that Test 2 is stronger than Test 1. Table 1 provides a comparison of the two tests for eliminating variables v_i^q in model FMCC3. The table only includes the instances in which reductions were observed.

In the range $Q=\{1,\ldots,q_{max}\}$, Test 1 is only effective on the very sparse instances, while Test 2 can reinforce the variables elimination on the same instances and still promote reductions on the c-fat class. As expected, the tests have no efficacy on the more dense graphs.

Instances	n	density	q_{max}	Total number of variable	Test 1		Test 2	
					nº of variables after elimination	% eliminated	nº of variables after elimination	% eliminated
c-fat200-1	200	0.077	12	2400	2400	0.00	1578	34.25
c-fat200-2	200	0.163	24	4800	4800	0.00	3246	32.38
c-fat200-5	200	0.426	58	11600	11600	0.00	8516	26.59
c-fat500-1	500	0.036	14	7000	7000	0.00	4640	33.71
c-fat500-2	500	0.073	26	13000	13000	0.00	9260	28.77
c-fat500-5	500	0.186	64	32000	32000	0.00	23254	27.33
c-fat500-10	500	0.374	126	63000	63000	0.00	46752	25.79
p_hat700-1	700	0.249	16	28000	28000	0.00	27994	0.02
p_hat700-2	700	0.498	60	59500	59500	0.00	59486	0.02
d1-RTN	2418	0.0032	10	24180	13675	43.44	7725	68.05
d3-RTN	4755	0.0024	18	85590	36281	57.61	21128	75.31
d7-RTN	6511	0.0021	18	117198	51664	55.92	31091	73.47
d15-RTN	7965	0.0020	22	175230	69769	60.18	42390	75.81
d30-RTN	10101	0.0018	27	272727	98362	63.93	60291	77.89
d66-RTN	13308	0.0017	36	479088	151839	68.31	94362	80.30

Table 1: Variables elimination tests effectiveness involving model FMCC3.

The variables' elimination test performed in Martins (2010), and addressing the MC problem, took into account the entire range $Q=\{1,\ldots,n\}$. For that reason, the results described in Martins (2010) are significantly different from those here reported.

3.2 Sequential algorithm for solving the FMCC3 model

We also propose a sequential approach for solving model FMCC3 that profits from the modular structure of the formulation. This is motivated by the fact that if we remove the single constraint (8) from formulation FMCC3 we get a separable model, with a separate formulation for each q-sized clique subproblem, for $q\in Q$. Considering this fact, and instead of solving each of the individual subproblems, we consider a partition of set Q into smaller subsets, each one characterizing a restricted MCC subproblem. Thus, suppose we have the following partition $Q=Q^1\cup Q^2\cup\ldots\cup Q^k$ and let FMCC3(Q^i) denote the FMCC3 model restricted to the range Q^i of clique's sizes, for $i=1,\ldots,k$. We also denote by $z(Q^i)$ the optimum solution value of FMCC3(Q^i), while q^i is the cardinality of the associated clique. Then, we propose the following algorithm.

Sequential algorithm for solving the FMCC3 model

1. Set $i\leftarrow 1$ and $z(Q^0)\leftarrow 0$;
2. Solve model FMCC3(Q^i), returning a clique with cardinality q^i ;

while (FMCC3(Q^i) is feasible and $i < k$) **do**
    $i \leftarrow i + 1$;
    Set $z(Q^{i-1})$ as a lower bound;
    Solve model FMCC3(Q^i), returning a clique with cardinality q^i (if existing);
end-while;
if (FMCC3(Q^i) is infeasible)
then return $z(Q^{i-1})$ and q^{i-1}.
else return $z(Q^i)$ and q^i.

The algorithm exits the while loop when FMCC3(Q^i) is infeasible, for a given iteration i, which usually happens before reaching the last set Q^k. However, it may go into set Q^k if FMCC3(Q^{k-1}) is still a feasible problem.

We can strengthen each subproblem FMCC3(Q^i) branch-and-bound process by supplying a given lower bound to the global optimum value. This can be done by using the previous iteration optimum solution value, given by $z(Q^{i-1})$.

Based on some empirical experiments, model FCCM3(Q^i) is easier when Q^i includes smaller cardinality values, becoming harder for the last subsets in the partition. For this reason, we chose to partition Q into decreasing sized subsets, that is, $|Q^1| \geq |Q^2| \geq \ldots \geq |Q^k|$. In fact, we have observed that the last FCCM3(Q^i) subproblems require much more time than the former. For these reasons, we have considered the following methodology for constructing each instance partition, using a given partition factor r, with $r \leq 1$. Each subset is represented by $Q^i = \{q^i_{\min} \ldots q^i_{\max}\}$.

Procedure for constructing the partition $Q^1 \cup Q^2 \cup \ldots \cup Q^k$, for a given factor r

Set $q^0_{\max} \leftarrow 0\; q^0{}_{\max} \leftarrow 0$;
for $i = 1,\ldots,k$ **do**
    $i \leftarrow i + 1$;
    Set $q^i_{\min} \leftarrow q^{i-1}_{\max} + 1$ and $q^i_{\max} \leftarrow q^{i-1}_{\max} + \left| \left(q_{\max} - q^{i-1}_{\max}\right)/r \right|$;
end-for;

Due to the decreasing factor r, the last subsets usually include a single element.

4. Computational tests addressing the MCC problem

In order to test the computational performance of the algorithms and before addressing the practical case on Market Basket Analysis, we provide some computational experiments using DIMACS benchmark instances.

All tests were conducted on an Intel Core i7-2600 with 3.40 GHz and 8 GB RAM. The experiments made on the heuristics were performed on Linux operating system. The algorithms were compiled with gfortran with flag –O2. User times for DIMACS machine benchmark instances are 0.00, 0.02, 0.14, 0.93 and 3.55 seconds, for instances `r100.5`, `r200.5`, `r300.5`, `r400.5` and `r500.5`, respectively (Johnson & Trick, 1996).

The tests conducted on the models reported in Section 2 were run on the same machine but under Windows 7 operating system. The formulations were solved using the IBM/ILOG/CPLEX 11.2 package. We used most default settings, which involve an automatic procedure that uses the best rule for variable selection and the best-bound search strategy for node selection in the branch-and-bound tree. The automatic generation of additional global cuts was closed, because in most cases it revealed to be more time demanding. We have set an upper time limit of 10800 seconds in all tests.

The selected instances from the DIMACS benchmark database were taken from the c-fat, p_hat, keller, C and MANN families. We also included strongly sparse graphs, using the Reuters terror news (RTN) instances proposed in Corman, Kuhn, McPhee & Dooney (2002) and made available in Pajek's data base (Batagelj & Mrvar). In our tests we considered the same subgraphs described in Martins (2012), addressing the observations collected during 1, 3, 7, 15, 30 and 66 days, corresponding to instances d1-RTN, d3-RTN, d7-RTN, d15-RTN, d30-RTN and d66-RTN, respectively.

4.1 Tests with the exact methods for the MCC problem

We start analyzing the results produced by the exact methods, described in Section 2.

Table 2 reports the results obtained with models FMCC1 and FMCC2, while Table 3 presents the results from model FMCC3. We considered two strategies for testing model FMCC3: i) in a full range version, using the full range set $Q=\{1,\ldots,q_{max}\}$; and ii) using the Sequential algorithm proposed in Subsection 3.2, considering three partitions for Q:

- Partition 1: is a full partition version, with $Q=Q^1\cup Q^2\cup\ldots\cup Q^k = \{1\}\cup\{2\}\cup\ldots\cup\{q_{max}\}$ and $k = q_{max}$, involving a partition factor $r = 1/_{qmax}$, being full granular;
- Partitions 2 and 3: are characterized by partition factors strictly greater than $1/q_{max}$, being less granular. We tested a number of factors, ranging from $r = 1/200$ to $r = 3/4$, and chose the two partitions with the best results (time and solution quality). Partition 3 is the less granular, having the largest partition factor.

Next we describe the notation used in Tables 2 and 3. Columns with labels “opt” and “time”, under “LP relaxation” give the linear programming (LP) relaxation optimum and the time to reach it, respectively. Columns “opt / ≥ best lb” under “Branch and bound” provide the integer optimum values or the best lower bounds to the optimums when the optimum is not attained within the given time limit (10800 seconds) or when CPLEX stops due to memory limitations (being denoted by “o-m” in the time’s column). In this case, the lower bound value is preceded by “≥”. Bold values in columns “opt / ≥ best lb” indicate the best solutions attained among all the models under discussion. Columns with label “time” under “Branch and bound” give the CPU times to reach the reported optimum/best solution values. In both tables, the two last columns, under “Best solution”, give the best solution value among all the models under discussion and the associated clique size, respectively. The upper bounds for the clique size (q_{max} values in Q) are given in column 5. These values were taken from (Gendron, et al., 2008) and (Martins, 2012). Times are reported in CPU seconds in all tables. Cells with no value (“---“) indicate that CPLEX was not able to read and preprocess the model.

In Table 3, we also include information concerning the partitions used within the Sequential algorithm, namely, the partition factor (r), the number of subsets in the partition (k) and the iteration in which the Sequential algorithm stopped (i). This information is given in the triplet (r, k, i), under the columns with label “(r, k, i)”.

(Please, include Table 2 about here)

Table 2: Computational results with the FMCC1 and FMCC2 models for the MCC problem.

(Please, include Table 3 about here)

Table 3: Computational results with the model FMCC3 for the MCC problem, with and without the Sequential algorithm.

As expected, the results reported in Tables 2 and 3 indicate that, in general, model FMCC3 is better suited for very sparse graphs (see the RTN class and some c-fat instances), while model FMCC1 performs better for very dense graphs (see the MANN class and some smaller sized p_hat instances). Model FMCC2 behaves better for the large sized mid-density graphs (see the larger density c-fat instances, the largest size p_hat instances, keller6, c1000_9 and c2000_9 instances). Furthermore, the full range version of FMCC3 has not been able to handle very dense graphs, although it reaches the optimums in the very large RTN class, to which the FMCC1 and FMCC2 models were only able to solve the smallest instance (d1-RTN). On the opposite side, model FMCC1 was the only one to solve most MANN class instances. Yet, when using the Sequential algorithm for handling model FMCC3, its performance is much improved, competing in a larger extent with formulations FMCC1 and FMCC2. This performance is more effective for the more granular partitions, namely Partition 1. In fact, these versions of model FMCC3 are truly much faster to reach the RTN optimums, and also faster to reach the p-hat and keller4 instances' optimums, when compared with the other formulations. It is also worth to note the efficacy of the versions involving Partitions 1 and 2 among a number of mid-density and large sized instances, namely among the p_hat700, p_hat1000, p_hat1500-1, c2000_5 and c4000_5 instances.

We also would like to stress the relevancy of using the previous iteration optimum value as a lower bound for the current iteration branch-and-bound execution, within the Sequential algorithm. This aspect is particularly relevant when solving the last subset, namely when it characterizes an infeasible problem. To exemplify this aspect, we have solved instance d66-RTN using model FMCC3 with the Sequential algorithm and Partition 1, but ignoring the mentioned use of the previous iteration optimum value. In this case, the algorithm took 4486.49 seconds to reach the optimum, where most of this time was consumed for solving the last iteration infeasible subproblem, requiring 4467.38 seconds.

Table 4 summarizes the performance of the models discussed in the present section, considering the number of instances to which each model found the best solution and the optimums. The total number of instances is 42, among which we have found 24 optimums.

	FMCC1	FMCC2	FMCC3			
			Full range	Partition 1	Partition 2	Partition 3
number of best solutions	22	24	16	29	28	23
number of optimums	16	14	15	21	21	19

Table 4: Number of best solutions and optimums.

The records in Table 4 confirm the advantage of model FMCC3 with the Sequential algorithm and using the full granular partition (Partition 1). This performance degrades when the partition becomes less granular, leading to poor results in the Full range version.

4.2 Tests with the heuristics for the MCC problem

Tables 5 and 6 report the computational results obtained with the R-ILS and D-ILS algorithms for the MCC problem. Table 5 experiments involve the original versions of the two algorithms, as described in (Grosso, et al., 2008) and designed for the MC problem. The results of the versions adapted for the MCC problem that use Proposition 1 for characterizing the sets $K_0(C)$ and $K_1(C)$, as described in conditions (1) and (2) in Section 2, are reported in Table 6. We set the maximum number of iterations parameter $max\_sel = 10^5$ for all c-fat, p_hat, C, keller and MANN instances and set $max\_sel = 10^3$ for the RTN class. Both algorithms were tested using 100 runs on each instance. Times are reported in CPU seconds.

We used the following notation in the tables: "best_sol" represents the best solution cut-clique cardinality, "$|C|$" is the best solution clique's cardinality, "sol_avg" represents the average cut-clique size over the 100 runs, "time_avg" is the average CPU time (in seconds) for a run, "#best" represents the number of best solutions during the 100 runs, "best_sel_avg" is the average number of iterations (selections) to reach the best solution (only among best solutions). Values in bold in columns "best_sol" indicate the best solutions found among the four algorithms.

(Please, include Table 5 about here)

Table 5: Results of the algorithms R-ILS and D-ILS without Proposition 1, for the MCC problem.

(Please, include Table 6 about here)

Table 6: Results of the algorithms R-ILS and D-ILS with Proposition 1, for the MCC problem.

Comparing the results in the two tables, we can state that Proposition 1 is only effective in the more rational degree based method (D-ILS). In effect, the two strategies together (Proposition 1 and degree based methodology) produce a more powerful scheme for addressing the MCC problem. We also note that the random based method (R-ILS) performs better without the limitations imposed by Proposition 1. Also, the computational effort for handling Proposition 1 imposes a slight increase on CPU time.

Some of these aspects are summarizes in Table 7, considering the results reported in Tables 5 and 6, namely the number of best solutions and the average time per run (average of the time_avg values) for the four algorithms under discussion.

	without Proposition 1		with Proposition 1	
	R-ILS	D-ILS	R-ILS	D-ILS
number of best solutions	36	37	34	41
average time per run	1.1623	1.4842	1.1830	1.4779

Table 7: Number of best solutions and average time per run, for the entire set of 42 instances.

Now, considering just the results that incorporate Proposition 1, reported in Table 6, and the summarized figures in Table 7, we can credit algorithm D-ILS as being more advantageous than R-ILS, because it was more assertive (see column #best in Table 6 and the number of best solutions in Table 7). In fact, the D-ILS only missed one best solution (keller6) among the entire set of instances, within the given *max-sel*=10^5 iterations limit. However, algorithm D-ILS usually took more time than R-ILS to solve each run (see "time_avg" columns in Table 6 and the average time per run in Table 7), although it reached the best solutions much sooner than algorithm R-ILS, in general (see "best_sel_avg" columns in Table 6), so could possibly require a smaller number of iterations than R-ILS. While the random based methodologies performed better in the experiments conducted in Grosso, et al. (2008) addressing the MC problem, when compared with those based on degree choices, our results indicate the contrary. In fact, this is not surprising, because the objective function of the MCC problem is strongly dependent on nodes' degrees conditions, which influence the strategy when searching the entire cliques' feasibility space. Actually, we also noticed that the random based algorithm was faster; however, and as mentioned above, the choice of nodes based on degree information is a crucial issue for conducting the search within the MCC problem feasibility space.

It is also worth to observe that most optimums obtained by the exact methods were also attained by the heuristics in all the 100 runs, namely by the D-ILS algorithm with Proposition 1. The only exceptions are the very dense MANN_a27 and MANN_a45 instances. In addition, and considering the non-optimum best solutions attained by the exact methods, the heuristics produced better results for more than 28% of the graphs (12 instances). However, for the very dense graphs MANN_a45 and MANN_a81, the heuristics' results were defeated by the best feasible solution values obtained with model FMCC1. As usual, the heuristics required much less CPU time, in general.

Considering the results reported in Table 6, involving the ILS algorithms that incorporate Proposition 1, and in order to improve the results of the keller, C and MANN instances that have failed reaching the best solution in all 100 runs, we increased the total number of iterations, considering *max-sel*=10^7, leading to the results reported in Table 8. As expected, these longer executions were more time demanding, but the number of successful solutions increased and the quality of the best solution value in keller6, c2000_9 and c4000_9 was also improved. Yet, the best solution values attained by model FMCC1 for the MANN_a45 and MANN_a81 instances were still unbeaten.

	R-ILS						D-ILS					
	best_ sol	|*C*|	sol_avg	time_ avg	#best	best_ sel_avg	best_ sol	|*C*|	sol_avg	time_ avg	#best	best_ sel_avg
keller5	**15184**	27	15184	88.042	100	6397.3	**15184**	27	15184	115.072	100	35577.1
keller6	**159608**	59	157898.20	383.806	5	4093735.1	**159608**	59	152436.41	513.088	2	4654674.5
c500_9	**22691**	57	22691	33.644	100	46322.8	**22691**	57	22691	49.137	100	116825.0
c1000_9	**57149**	68	57148.61	117.952	87	2532921.3	**57149**	68	57148.73	166.388	91	3700439.2
c2000_5	16097	16	16081.48	403.816	5	4574118.0	**16106**	16	16105.10	463.515	93	2143196.9
c2000_9	135060	78	133289.35	257.996	3	5192927.5	**136769**	79	135490.18	352.743	1	5197927.4
c4000_5	36101	18	35777.92	936.199	1	3638990.7	**36174**	18	35935.71	1101.315	22	5697591.6
MANN_a27	**31284**	126	31284	11.205	100	88284.2	**31284**	126	31284	29.021	100	100615.7
MANN_a45	235422	341	235036.36	93.572	10	3478494.9	**236406**	344	235832.16	169.821	3	55377.1
MANN_a81	2427048	1089	2424869.3	353.782	8	3631875.4	**2436894**	1098	2434963.5	693.297	5	1939.5

Table 8: Heuristic results with *max-sel*=10^7, considering the R-ILS and D-ILS algorithms with Proposition 1.

Some tests were also conducted using alternative restart rules, namely those suggested in Grosso, et al. (2008). However, in our case, instead of perturbing the clique (C) returned from the previous stage by inserting an additional node, we chose to directly remove max(2,$|C|$/3) nodes from C, taken at random. Yet, although this strategy performed better than those used in Grosso, et al. (2008), the results didn't improve, in general, those reported in Table 6. For this reason, we opted not to include them in the paper.

5. Maximum cut-cliques applied to a market network

We have applied the proposed methods to a database of transaction data collected with home scanners.[1] We use a household panel database for the British ice cream market containing information over a 2½-year period (January 2006 to June 2008) among 142 households. The dataset includes information for a total of 4,899 items purchased during the period studied, chosen from a total of 691 different *varieties of products* (SKUs)[2] available in the British market. Considering the database, let us define a basket as the set of products purchased on the same day and in the same shop by a household. We form a basket for each group of products that have the same household (*houseid)*, purchase date (*purdate),* and shop (*shopid)*. For instance, the following two products constitute a basket:

househid	***purdate***	***shopid***	***prod***	***Desc***
10043	17002	770	52377	SAIN VNLLA 1LT
10043	17002	770	628805	WLL CRT DOR GREEK YGT+HNY 900ML

[1] See Brennenberg et al. (2008) for a detailed description of IRI home scanner database.
[2] Stock Keeping Unit.

The dataset contains a total of 2,993 baskets, averaging 1.64 ice cream products (SKUs) per basket. The transactions were made by individual clients, but we have not associated each transaction to a specific customer due to privacy concerns. The analysis has been conducted at the household level, regardless of the family member who made the purchase. The purchase frequency varies significantly across households. The one with the largest number of purchases is household "10918" with 324 products bought within a total of 244 baskets. The number of purchases also varies significantly by product. The one bought most frequently is product *Walls Magnum Stick (3-unit package)*. A total of 133 units of this product were bought by 34 households in 93 different baskets during the entire period.

The number of products (SKUs) in the network is 691, and the number of edges is 1181. If two products (*i* and *j*) are found in the same basket, then the network includes edge (*i,j*), otherwise, if there is no basket with products *i* and *j*, then edge (*i,j*) doesn't belong to the network. In addition, there are 177 isolated nodes (products) in the network, with degree 0. These products never share the same basket with any other product in the network.

All computational experiments here described were run under the same machine conditions as those reported in Section 4. Like before, each heuristic algorithm was tested using 100 runs on each instance.

We have determined the maximum clique and the maximum cut-clique. We now describe the results, the implications for the analysis of the product interactions, and the managerial implications. The largest clique in the network (maximum clique) includes the following 8 products (see Figure 3):

prod id number	prod description		# external links
147	FRDRKS DARK CHOC ICE VNLA	10PK	4
148	FRDRKS CHOC ICE NPLTN	10PK	13
149	FRDRKS LGHT CHOC ICE VNLA	10PK	3
375	CDBRY CONE DRY MLK MINT	4PK	9
489	DLMNT LLY RSPBRY SMOOTHIE	3PK	16
518	NSTL LOLLY ROLO STCK	6PK	0
539	FRDRKS CHOCOLATE	2LT	0
541	FRDRKS STWBRY	2LT	0

We used one of the formulations described in Martins (2010) for the maximum clique problem.

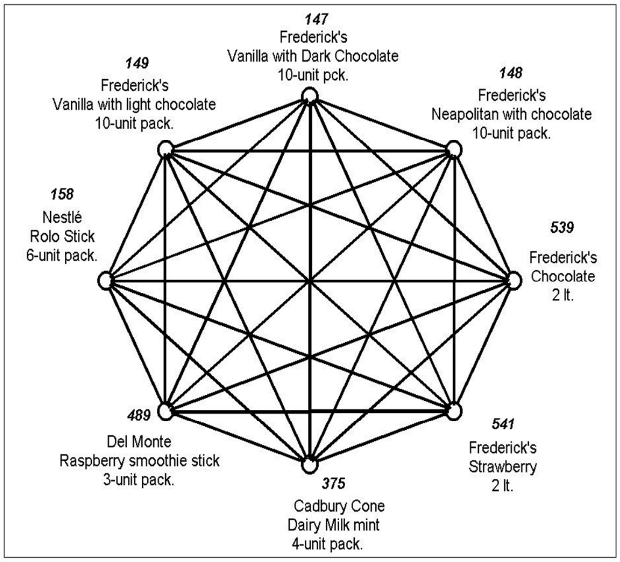


Figure 3: Largest cardinality clique.

This solution indicates that there are baskets with all possible pairings among these 8 products. This is the largest set of products having this property. Together, these products have 45 links to the remaining products in the network, which means that, apart from the clique's baskets, each one of the 8 products share the same basket with other products in 45 occasions. Those other adjacent products involve 32 items.

The maximum-clique analysis allows for the identification of the biggest set of items purchased in conjunction with all others at least once in a common basket. The analysis of the product network reveals some interesting results. As explained previously, the cliques constitute groups of products that have been pairwise bought together in a basket, among all the products in the set. As for many product categories, a significant segment of households buy more than one ice cream in the same basket. Identification of cliques allows for determining the attributes or dimensions in which multiple-purchase households seek variety. For instance, five out of the eight products constituting the maximum clique are big formats - 2 liters or 10-unit packages - of an Italian-style luxury brand, Fredericks's, varying in the following five flavors: Chocolate, Strawberry, Vanilla ice cream with dark chocolate, Vanilla ice cream with light chocolate, and Neapolitan ice cream with chocolate. The remaining three products of the basket are also multiple-unit packages of three products from leading manufacturer brands: Del Monte, Nestlé, and Cadbury.

Products in the maximum clique set have the common characteristic of being bought together by variety-seeking households. However, this is not a purchase pattern for all products, as 26% of the products available were never bought together with any other ice cream in the whole network. For instance, two well known distributor-brand products, "Tesco's vanilla ice cream with light chocolate" and "Sainsbury's Morrison's vanilla ice cream with raspberry", were purchased 8 times and 13 times respectively, always in single-product baskets.

Noteworthy, the set of products constituting the maximum clique is not the clique with largest incidence to other products in the network. In fact, the set of products forming a clique and with the largest number of links to the remaining products in the network (maximum cut-clique) involves only 6 products, although it has 100 links to the remaining products. Those 6 products are (see Figure 4):

prod id number	prod description		# external links
21	WALLS BL RBN VNLLA	2LT	10
65	WALLS MINI MILK LOLLIES	12PK	14
66	MARS CHOC ICE	4PK	23
72	WALLS MAGNUM WHITE STCK	3PK	23
80	NSTL LOLLY FAB	8PK	21
305	WALLS MNI TWISTER STW+LMN	8PK	9

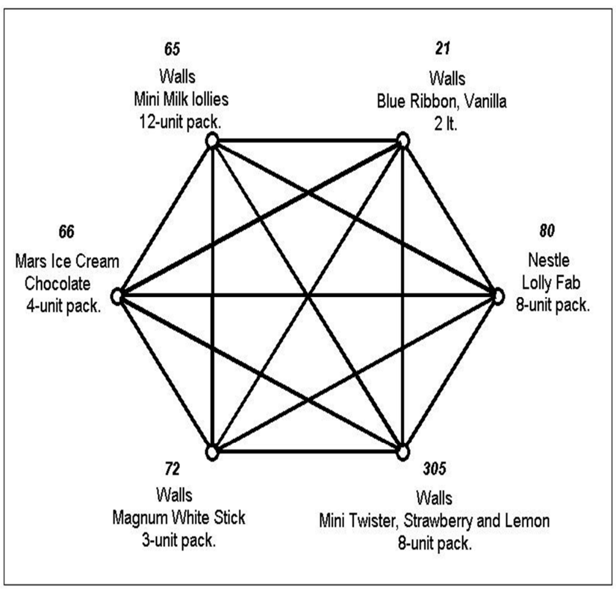


Figure 4: Clique with maximum neighborhood.

These 6 products form a clique, which means that all pairs of products in this set are found in householder's baskets. They are adjacent to 75 other products in the network, involving 100 links, which stresses their strong engagement. So, probably, most of the householders buying these 6 products are also strong potential buyers for the remaining products, especially those products involved in the 100 links.

This solution took 0.03 seconds to be reached, using an exact model and less than 0.005 seconds using a heuristic. Tables 9 and 10 show the results from the heuristic and exact methodologies proposed in Sections 2 and 3, for the market network under discussion. We use the same notation considered in Tables 2, 3 and 6. Parameter $q_{max} = 9$.

FMCC1				FMCC2				FMCC3			
LP relaxation		Branch and bound		LP relaxation		Branch and bound		LP relaxation		Branch and bound	
opt	time	opt	time	opt	time	opt	time	opt	time	opt	time
136.0000	0.34	100	1.43	165.6635	0.05	100	0.70	111.4014	< 0.01	100	0.03

Table 9: Exact formulations' results for the market network under discussion.

R-ILS						D-ILS					
best_sol	$\|C\|$	sol_avg	time_avg	#best	best_sel_avg	best_sol	$\|C\|$	sol_avg	time_avg	#best	best_sel_avg
100	6	99.51	0.0132	93	249	100	6	100	0.0048	100	45

Table 10: Heuristic algorithms' results for the market network under discussion.

Once again, formulation FMCC3 was the fastest to reach the optimum, confirming its advantage on sparse networks, as observed in Subsection 4.1. Likewise, and as concluded in Subsection 4.2, the algorithm D-ILS was the most accurate to find the best solutions, and once again, it was faster than the R-ILS, as observed with the sparser RTL class instances (see, Table 6).

An alternative solution, with a smaller number of links (just 93) is:

```
prod id number    prod description                       # external links
      21          WALLS BL RBN VNLLA         2LT                 10
      72          WALLS MAGNUM WHITE STCK    3PK                 23
      80          NSTL LOLLY FAB             8PK                 21
     190          WALLS CORNETTO STWBRY      6PK                 20
     192          WALLS CORNETTO CHOC N NUT  6PK                 10
     305          WALLS MNI TWISTER STW+LMN  8PK                  9
```

In both solutions, the maximum cut-clique returned smaller sets, compared to the maximum clique. However, we can also see that in both solutions the set of products obtained, when maximizing the external interactions is formed by the top-selling ice creams from leading brands: Cornettos (Walls), Magnums (Walls), Blue Ribbon (Walls), Lolly fabs (Nestlé) and Mars chocolate bar (Mars).

Figure 5 compares the neighborhoods of the previously discussed cliques.

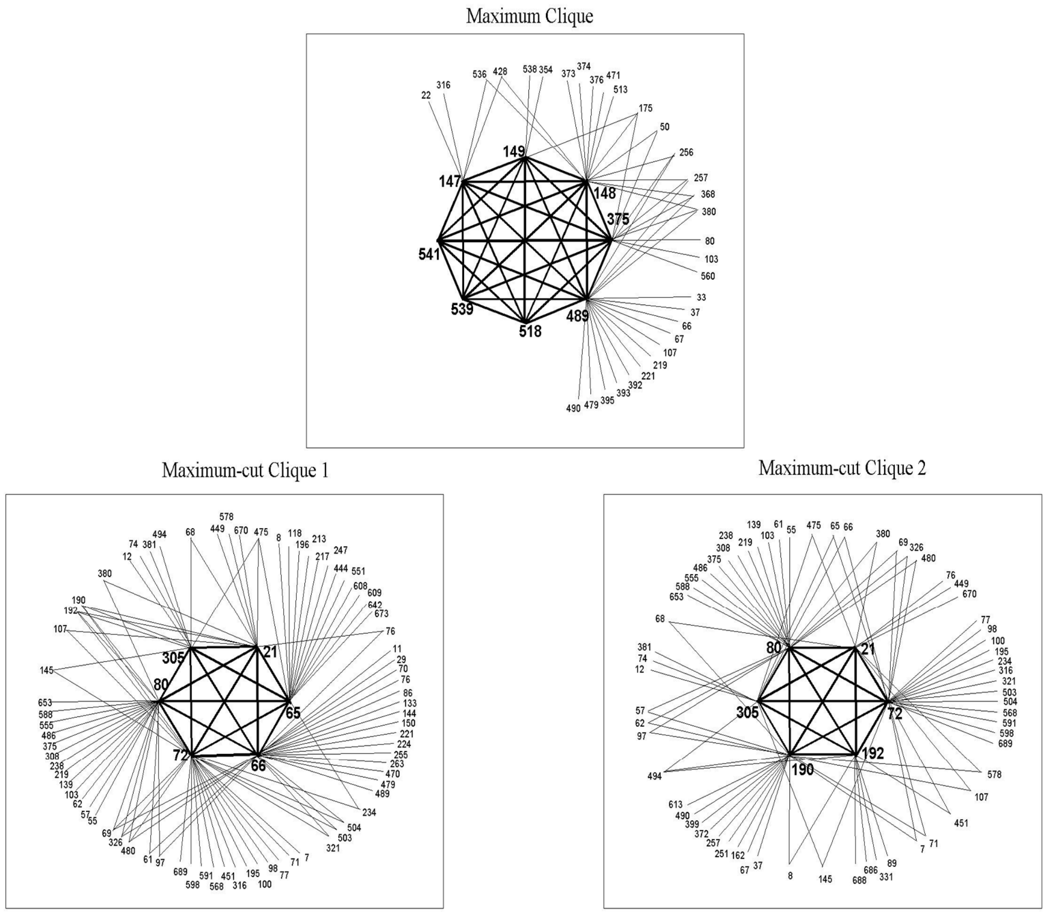


Figure 5: Comparing the neighborhoods of the three cliques.

Noteworthy, while the maximum-clique analysis focuses on the biggest set of interacting products independent of the number of purchases, some items in the set may have been bought just a few times, or even only once. Therefore, the maximum-clique analysis may not be the most appropriate technique to identify and analyze representative product interactions in the dataset. However, the maximum cut-clique analysis identifies the set of products with the maximum number of links to the other products of the network, revealing interacting patterns from leading-sale products. A visual comparison among the two largest cut-cliques and the maximum clique reveals significant differences among the selected products in the three cliques.

In what follows we extend the analysis by including weights on the edges of the market network under discussion, with w_{ij} representing the number of times that products i and j were bought together, for each edge $(i,j)\in E$, then we can think about using the weighted versions of the MC and the MCC problems, namely, the maximum edge-weight clique (MEWC) and the maximum edge-weight neighborhood clique (MEWNC) problems, respectively. Hence, using the formulations proposed in Gouveia & Martins (2013), the MEWC optimum solution includes 7 nodes, with no common products with the previously discussed solutions, while the MEWNC optimum is entirely equal to the maximum cut-clique 1 solution. Noteworthy, while the maximum-clique solution is sensitive to the weighted approach, the maximum-cut-clique solution has shown to be independent on the weights, which brings robustness to the analysis of the basket with the maximum external influence. The 6 products in the MEWC optimum are:

```
prod id number    prod description                     # external links
     367         WAIT SORBET MANGO        750ML              11
     393         YRKSHRE DLES DRY STCKY TFFE1LT              18
     395         OB CHOC ICE VNLLA         10PK              23
     399         OB CHOC STKS VNLLA         3PK              10
     496         M+MS PRM ICE CRM         500ML               6
     591         ST M ORGNC SORBET RSPBRY 500ML               7
     600         TSC MMMM LOLLY WHT CHOC    3PK               4
```

Once again, if we compare the selected products in the maximum clique solutions (from the MC and the MEWC problems) with those from the cut-clique versions, we can affirm that the products in the cut-clique solutions are more influential than those from MC and MEWC.

Another relevant aspect to observe involves the average number of purchases per product in the various solutions under discussion. This result can be calculated considering the total number of purchases among the products in the clique (using the weights on the edges in the clique), or considering the total number of purchases among clique and non-clique products (using the weights on the edges in the cutset). Table 11 summarizes these results, including descriptive information about the cliques and their neighborhoods.

Solution	Number of			Average number of purchases per product	
	products	external products	external links	in the clique	in the neighborhood
Maximum-clique solutions					
Maximum clique	8	32	45	4.6	7.5
Maximum edge-weight clique	7	47	79	6.0	13.6
Maximum cut-clique solutions					
Maximum cut-clique 1	6	75	100	5.2	26.5
Maximum cut-clique 2	6	63	93	5.3	24.3

Table 11: Descriptive values for the maximum clique, the maximum cut-clique and the maximum edge-weight clique solutions.

These results show that the cut-cliques' selections are more effective, attracting many more products, on average, than the other solutions. In effect, the maximum cut-clique solutions may possibly capture a central subset of elements that may represent key players in the entire environmental system.

6. Conclusions

In this work we consider the Maximum Cut-Clique problem, using three mathematical models and proposing Iterated Local Search heuristics to solve it (R-ILS and D-ILS algorithms). We also describe the importance of the application of cliques in market networks, addressing marketing related problems in the field of Market Basket Analysis. The Maximum Cut-Clique can identify relevant relationships among products in a market basket and be complementary to other MBA approaches, as exemplified with a database for the British ice cream market.

We proposed a new methodology that explores the discretized property of a known formulation for the MCC problem. The new methodology produced very competitive results on most instances tested, being particularly efficient among the sparser graphs.

We have also performed computational experiment on the R-ILS and D-ILS algorithms for the MCC problem, generating the first lower bounding results for some DIMACS benchmark instances, including the largest graphs. Most of these lower bounds were confirmed to be the optimums, and in some cases, the heuristics' results outperformed the best solutions (non-optimums) returned by the exact methods, namely among the larger instances in the p_hat, keller and C classes.

We also presented the application of the algorithms to a household panel database for the British ice cream market. The results revealed the importance of applying these techniques to obtain relevant information from consumer's databases.

This work opens a new line of research related with the application of Clique based models to evaluate market baskets and obtain a different type of information, when compared to the traditional approaches in marketing. Future work is oriented to the application of these models and techniques to larger databases on several different products and markets. An important extension is to consider a network with weights on the edges, and look for the maximum weight cut-clique.

Acknowledgments

This work has been partially supported by the Portuguese National Funding by FCT (project PEst-OE/MAT/UI0152), Pedro Martins; and by the Ministerio de Economía y Competitividad (Spanish Government), A. Ladrón-de-Guevara (ECO2011-27942) and H. Ramalhinho (TRA2013-48180-C3-2-P). We also thank the referees for valuable comments and suggestions and thank Teresa Lameiras for an English revision of the text.

References

Aguinis, H., Forcum, L. E., & Joo, H. (2012). Using Market Basket Analysis in Management Research. *Journal of Management*, *39*(7), 1799–1824.

Batagelj, V., Mrvar, A. http://vlado.fmf.uni-lj.si/pub/networks/pajek/. Accessed in August, 2009.

Battiti, R., Protasi, M., 2001. Reactive local search for the maximum clique problem. *Algorithmica* 29, 610-637.

Bellare, M., Goldreich, O., Sudan, M., 1998. Free bits, PCPs, and nonapproximability - Towards tight results. *SIAM Journal on Computing* 27, 804-915.

Bomze, I.M., Budinich, M., Pardalos, P.M., Pelillo, M., 1999. The maximum clique problem. In: Du, D.Z., Pardalos, P.M. (Eds.), *Handbook of Combinatorial Optimization (vol. A)*. Kluwer Academic Publisher, Dordrecht. pp. 1-74.

Bull, S.C., Muldoon, M.R., Doig, A.J., 2013. Maximising the size of non-redundant protein datasets using graph theory. *PloS one* 8(2), e55484.

Cavique, L., 2007. A scalable algorithm for the market basket analysis. *Journal of Retailing and Consumer Services* 14, 400-407.

Corman, S.R., Kuhn, T., McPhee, R., Dooney, K., 2002. Studying complex discursive systems: Centering resonance analysis of organizational communication. *Human Communication Research* 28(2), 157-206.

Crescenzi, P., Fiorini, C., Silvestri, R., 1991. A note on the approximation of the max clique problem. *Information Processing Letters* 40, 1-5.

Della Croce, F., Tadei, R., 1994. A Multi-KP Modeling for the maximum-clique problem. *European Journal of Operational Research* 73, 555-561.

Gendron, B., Hertz, A., St-Louis, P., 2008. A sequential elimination algorithm for computing bounds on the clique. *Discrete Optimization* 5, 615-628.

Gouveia, L., Martins, P., 2013. Solving the maximum edge-weight clique problem in sparse graphs with compact formulations. CIO – Working Paper 3/2013.

Grosso, A., Locatelli, M., Pullan, W., 2008. Simple ingredients leading to very efficient heuristics for the maximum clique problem. *Journal of Heuristics* 14, 587-612.

Hipp, J., Günther, U., Nakhaeizadeh, G., 2000. Algorithms for association rule mining – a general survey and comparison. *ACM SIGKDD Exploration Newsletter* 2, 58-64.

Johnson, D.S., Trick, M., 1996. *Cliques, coloring, and satisfiability: Second DIMACS Implementation Challenge* (Vol. 26). American Mathematical Society, Providence, RI.

Karp, R.M., 1972. Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W. (Eds.), *Complexity of computer computations*. Plenum Press, New York. pp. 85-103.

Kamakura, W. A. (2012). Sequential market basket analysis. *Marketing Letters*, *23*(3), 505–516.

Keshavamurthy, B. N., Khan, A. M., & Toshniwal, D. (2013). Privacy preserving association rule mining over distributed databases using genetic algorithm. *Neural Computing and Applications*, *22*(S1), 351–364. Kim, H. K., Kim, J. K., & Chen, Q. Y. (2012). A product network analysis for extending the market basket analysis. *Expert Systems with Applications*, *39*(8), 7403–7410.

Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., & Verkamo, A. (1994). Finding interesting rules from large sets of discovered association rules. In *Proceedings of CIKM*, 401–407.

Lourenço, H.R., Martin, O., Stützle, T., 2003. Iterated local search. In: Glover, F., Kochenberger, G. (Eds.), *Handbook of Metaheuristics* (Vol. 57). Kluwer Academic Publishers, Norwell, MA. pp. 321-353.

Lourenço, H.R., Martin, O., Stützle, T., 2010. Iterated Local Search: Framework and Applications. In: Gendreau, M., Potvin, J.Y. (Eds.), *Handbook in Metaheuristics* (2nd. ed., Vol. 146). Springer. pp. 363-397.

Luz, C.J., Schrijver, A., 2005. A convex quadratic characterization of the Lovasz theta number. *SIAM Journal on Discrete Mathematics* 19, 382-387.

Martins, P., 2010. Extended and discretized formulations for the maximum clique problem. *Computers & Operations Research* 37, 1348-1358.

Martins, P., 2012. Cliques with maximum/minimum edge neighborhood and neighborhood density. *Computers & Operations Research* 39, 594-608.

Östergard, P.R.J., 2002. A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics* 120, 197-207.

Pullan, W., Hoos, H.H., 2006. Dynamic local search for the maximum clique problem. *Journal of Artificial Intelligence Research* 25, 159-185.

Raeder, T., Chawla, N.V., 2011. Market basket analysis with networks. *Social Network Analysis and Mining* 1, 97-113.

Raeder, T., & Chawla, N. (2009). Modeling a Store's Product Space as a Social Network. In *International conference on advances in social network analysis and mining, IEEE*, 164–169.

Solnon, C., Fenet, S., 2006. A study of ACO capabilities for solving the maximum clique problem. *Journal of Heuristics* 12(3), 155-180.

Spirin, V., Mirny, L.A., 2003. Protein complexes and functional modules in molecular networks. *Proceedings of the National Academy of Sciences* 100(21), 12123-12128.

Strickland, D.M., Barnes, E., Sokol, J.S., 2005. Optimal protein structure alignment using maximum cliques. *Operations Research* 53, 389-402.

Tomita, E., Kameda, T., 2007. An efficient branch-and-bound algorithm for finding a maximum clique. *Journal of Global Optimization* 37, 95-111.

Videla-Cavieres, I. F., & Ríos, S. a. (2014). Extending market basket analysis with graph mining techniques: A real case. *Expert Systems with Applications*, *41*(4), 1928–1936.

Zaki, M.J., 1999. Parallel and distributed association mining: A survey. *IEEE Concurrency* 7, 14-25.

Instances	n	density	$\omega(G)$	q_{max}	FMCC1 LP relaxation opt	FMCC1 LP relaxation time	FMCC1 Branch and bound opt / ≥ best lb	FMCC1 Branch and bound time	FMCC2 LP relaxation opt	FMCC2 LP relaxation time	FMCC2 Branch and bound opt / ≥ best lb	FMCC2 Branch and bound time	Best solution value	Best solution clique size
c-fat200-1	200	0.077	12	12	81.000	0.05	**81**	0.05	81.000	0.02	**81**	0.05	**81**	9
c-fat200-2	200	0.163	24	24	306.000	0.06	**306**	0.09	306.000	0.02	**306**	0.03	**306**	17
c-fat200-5	200	0.426	58	58	1892.000	0.05	**1892**	0.05	1892.000	0.00	**1892**	0.02	**1892**	43
c-fat500-1	500	0.036	14	14	110.000	0.66	**110**	0.76	110.000	0.05	**110**	0.23	**110**	10
c-fat500-2	500	0.073	26	26	380.000	0.64	**380**	0.80	380.000	0.03	**380**	0.23	**380**	19
c-fat500-5	500	0.186	64	64	2304.000	0.75	**2304**	0.83	2304.000	0.03	**2304**	0.16	**2304**	48
c-fat500-10	500	0.374	126	126	8930.000	0.52	**8930**	0.58	8930.000	0.03	**8930**	0.12	**8930**	94
p_hat300-1	300	0.244	8	9	954.000	0.11	**789**	34.73	984.077	0.01	**789**	4.99	**789**	8
p_hat300-2	300	0.489	25	29	5218.500	0.08	**4637**	255.67	5398.621	0.02	**4637**	4050.29	**4637**	25
p_hat300-3	300	0.744	36	51	10035.800	0.01	**≥ 7740**	10800	10379.354	0.02	≥ 7438	o-m	7740	36
p_hat500-1	500	0.253	9	13	2347.500	0.47	**1621**	3227.64	2403.868	0.03	**1621**	251.16	**1621**	9
p_hat500-2	500	0.505	36	46	14204.500	0.30	≥ 11333	o-m	14688.375	0.03	**≥ 11539**	o-m	11539	36
p_hat500-3	500	0.752	50	78	26675.000	0.34	**≥ 18859**	10800	27415.492	0.03	≥ 18305	o-m	18859	50
p_hat700-1	700	0.249	11	16	8794.000	2.01	≥ 2304	o-m	9084.689	0.08	**2606**	2950.82	**2606**	11
p_hat700-2	700	0.498	44	60	34082.500	1.12	≥ 19757	o-m	35913.813	0.05	≥ 19359	o-m	20078	43
p_hat700-3	700	0.748	62	102	61088.333	0.92	≥ 32675	10800	64239.966	0.05	≥ 32228	o-m	33057	61
p_hat1000-1	1000	0.245	≥ 10	20	16065.000	7.99	≥ 3278	o-m	16562.753	0.16	≥ 3385	10800	**3556**	10
p_hat1000-2	1000	0.490	≥ 46	76	62869.500	5.23	≥ 28893	o-m	66070.766	0.11	**≥ 30657**	o-m	30657	45
p_hat1000-3	1000	0.744	≥ 68	134	118907.000	3.09	≥ 40814	10800	123340.573	0.08	**≥ 48894**	o-m	48894	61
p_hat1500-1	1500	0.253	12	28	35502.500	33.21	≥ 4946	o-m	36363.071	0.36	≥ 5923	10800	**6018**	11
p_hat1500-2	1500	0.506	≥ 65	113	141600.500	15.18	≥ 49205	o-m	147468.755	0.28	**≥ 67486**	10800	67486	65
p_hat1500-3	1500	0.754	≥ 94	195	264945.000	10.76	≥ 80610	10800	273833.158	0.20	**≥ 111983**	10800	111983	93
keller4	171	0.649	11	17	1812.000	0.05	**1140**	17.05	1836.000	0.00	**1140**	95.57	**1140**	11
keller5	776	0.752	27	49	28046.000	0.08	**≥ 14760**	10800	28832.179	0.03	≥ 13288	o-m	14760	26
keller6	3361	0.818	≥ 59	122	336256.000	114.68	≥ 105984	o-m	339940.917	0.34	**≥ 136946**	10800	136946	50
c125_9	125	0.899	34	44	3094.500	0.00	**2766**	1.69	3150.306	0.00	**≥ 2766**	o-m	**2766**	34
c250_9	250	0.899	44	78	11680.000	0.02	**≥ 8123**	10800	11801.196	0.02	**≥ 8123**	o-m	8123	44
c500_9	500	0.901	≥ 57	144	44777.500	0.03	**≥ 22023**	10800	45094.729	0.03	≥ 20728	o-m	22023	55
c1000_9	1000	0.901	≥ 68	266	170838.000	0.22	≥ 46055	10800	171616.779	0.14	**≥ 47098**	o-m	47098	55
c2000_5	2000	0.500	≥ 16	110	102128.000	53.74	≥ 12191	o-m	102811.647	0.44	≥ 14180	10800	14344	14
c2000_9	2000	0.900	≥ 80	492	649108.500	9.39	≥ 91790	10800	651142.036	0.91	**≥ 110318**	10800	110318	63
c4000_5	4000	0.500	≥ 18	200	371095.000	497.83	---	---	372789.891	2.31	≥ 26437	o-m	28642	14
MANN_a9	45	0.927	16	18	426.000	0.00	**412**	0.00	426	0.00	**412**	0.25	**412**	16
MANN_a27	378	0.990	126	137	32220.000	0.01	**31284**	1.34	32606.000	0.00	≥ 31054	o-m	**31284**	126
MANN_a45	1035	0.996	345	367	241350.000	0.05	**236730**	106.07	244055.000	0.02	≥ 232362	o-m	**236730**	345
MANN_a81	3321	0.999	≥ 1100	1146	2474658.000	0.25	**≥ 2437978**	10800	2489112.000	0.03	≥ 2417040	o-m	2437978	1099
d1-RTN	2418	0.0032	10	10	1374.000	150.70	**1273**	162.87	1517.848	1.22	**1273**	15.58	**1273**	8
d3-RTN	4755	0.0024	18	18	---	---	---	---	4785.465	5.13	≥ 3460	o-m	**3526**	12
d7-RTN	6511	0.0021	18	18	---	---	---	---	---	---	---	---	**5656**	15
d15-RTN	7965	0.0020	18	22	---	---	---	---	---	---	---	---	**7772**	16
d30-RTN	10101	0.0018	21	27	---	---	---	---	---	---	---	---	**13099**	21
d66-RTN	13308	0.0017	28	36	---	---	---	---	---	---	---	---	**22379**	28

Table 2: Computational results with the FMCC1 and, FMCC2 models for the MCC problem.

FMCC3					Without the Sequential algorithm (full range)				With the Sequential algorithm									Best solution	
Instances	n	density	$\omega(G)$	q_{max}	LP relaxation		Branch and bound		Partition 1:	Branch and bound		Partition 2:	Branch and bound		Partition 3:	Branch and bound		value	clique size
					opt	time	opt / ≥ best lb	time	(r, k, i)	opt / ≥ best lb	time	(r, k, i)	opt / ≥ best lb	time	(r, k, i)	opt / ≥ best lb	time		
c-fat200-1	200	0.077	12	12	81.000	0.00	**81**	0.02	$(1/q_{max}, 12, 10)$	**81**	0.02	(1/10, 11, 9)	**81**	0.02	(3/4, 2, 2)	**81**	0.01	**81**	9
c-fat200-2	200	0.163	24	24	306.000	0.03	**306**	0.09	$(1/q_{max}, 24, 18)$	**306**	0.05	(1/20, 22, 16)	**306**	0.05	(1/2, 5, 3)	**306**	0.07	**306**	17
c-fat200-5	200	0.426	58	58	1892.000	0.22	**1892**	0.61	$(1/q_{max}, 58, 44)$	**1892**	0.37	(1/50, 54, 40)	**1892**	0.32	(1/10, 23, 12)	**1892**	0.50	**1892**	43
c-fat500-1	500	0.036	14	14	110.000	0.02	**110**	0.08	$(1/q_{max}, 14, 11)$	**110**	0.03	(1/2, 4, 3)	**110**	0.08	(3/4, 2, 2)	**110**	0.08	**110**	10
c-fat500-2	500	0.073	26	26	380.000	0.11	**380**	0.30	$(1/q_{max}, 26, 20)$	**380**	0.17	(1/10, 17, 11)	**380**	0.37	(1/5, 11, 6)	**380**	0.30	**380**	19
c-fat500-5	500	0.186	64	64	2304.000	0.67	**2304**	1.90	$(1/q_{max}, 64, 49)$	**2304**	0.99	(1/20, 39, 22)	**2304**	1.69	(2/3, 3, 3)	**2304**	1.76	**2304**	48
c-fat500-10	500	0.374	126	126	8930.000	3.84	**8930**	9.03	$(1/q_{max}, 126, 95)$	**8930**	4.45	(1/50, 83, 52)	**8930**	6.03	(1/20, 49, 25)	**8930**	6.44	**8930**	94
p_hat300-1	300	0.244	8	9	967.165	0.03	**789**	32.39	$(1/q_{max}, 9, 9)$	**789**	2.27	(1/5, 7, 7)	**789**	2.32	(1/2, 4, 4)	**789**	2.41	**789**	8
p_hat300-2	300	0.489	25	29	5390.571	0.36	≥ 229	o-m	$(1/q_{max}, 29, 26)$	**4637**	128.55	(1/10, 18, 15)	**4637**	128.76	(1/5, 11, 8)	**4637**	147.76	**4637**	25
p_hat300-3	300	0.744	36	51	10388.059	1.14	≥ 267	o-m	$(1/q_{max}, 51, 36)$	≥ 7587	o-m	(1/20, 33, 18)	≥ 7587	o-m	(1/10, 22, 10)	≥ 7438	o-m	7740	36
p_hat500-1	500	0.253	9	13	2365.519	0.17	**1621**	3796.91	$(1/q_{max}, 13, 10)$	**1621**	7.22	(1/5, 8, 5)	**1621**	7.40	(1/2, 4, 3)	**1621**	267.07	**1621**	9
p_hat500-2	500	0.505	36	46	14675.660	1.84	≥ 389	o-m	$(1/q_{max}, 46, 37)$	**≥ 11539**	10800	(1/20, 32, 23)	**≥ 11539**	10800	(1/10, 21, 12)	≥ 11333	10800	11539	36
p_hat500-3	500	0.752	50	78	27424.555	6.04	≥ 452	o-m	$(1/q_{max}, 78, 49)$	≥ 18305	10800	(1/50, 64, 35)	≥ 18305	10800	(1/20, 41, 17)	≥ 18305	10800	18859	50
p_hat700-1	700	0.249	11	16	8819.206	1.50	≥ 426	o-m	$(1/q_{max}, 16, 12)$	**2606**	126.14	(1/5, 9, 5)	**2606**	126.31	(1/2, 5, 3)	**2606**	2066.05	**2606**	11
p_hat700-2	700	0.498	44	60	35869.464	9.16	≥ 539	o-m	$(1/q_{max}, 60, 44)$	**≥ 20078**	10800	(1/20, 36, 20)	**≥ 20078**	10800	(1/10, 24, 11)	≥ 19757	10800	20078	43
p_hat700-3	700	0.748	62	102	---	---	---	---	$(1/q_{max}, 102, 61)$	**≥ 33057**	10800	(1/50, 75, 35)	**≥ 33057**	10800	(1/20, 45, 15)	≥ 31308	10800	33057	61
p_hat1000-1	1000	0.245	≥ 10	20	16082.792	4.68	≥ 408	o-m	$(1/q_{max}, 20, 11)$	**3556**	279.83	(1/10, 16, 6)	**3556**	279.83	(1/5, 10, 4)	**≥ 3556**	10800	**3556**	10
p_hat1000-2	1000	0.490	≥ 46	76	---	---	---	---	$(1/q_{max}, 76, 46)$	**≥ 30657**	10800	(1/50, 63, 33)	**≥ 30657**	10800	(1/20, 40, 15)	**≥ 30657**	10800	30657	45
p_hat1000-3	1000	0.744	≥ 68	134	---	---	---	---	$(1/q_{max}, 134, 62)$	**≥ 48894**	10800	(1/100, 117, 45)	**≥ 48894**	10800	(1/50, 86, 25)	≥ 48184	10800	48894	61
p_hat1500-1	1500	0.253	12	28	---	---	---	---	$(1/q_{max}, 28, 12)$	**6018**	7014.07	(1/20, 24, 8)	**6018**	7015.08	(1/10, 17, 5)	**≥ 6018**	10800	**6018**	11
p_hat1500-2	1500	0.506	≥ 65	113	---	---	---	---	$(1/q_{max}, 113, 59)$	≥ 62108	10800	(1/50, 79, 27)	≥ 60206	10800	(1/20, 48, 13)	≥ 62108	10800	67486	65
p_hat1500-3	1500	0.754	≥ 94	195	---	---	---	---	$(1/q_{max}, 195, 73)$	≥ 90095	10800	(1/100, 147, 36)	≥ 88955	10800	(1/50, 102, 20)	≥ 88955	10800	111983	93
keller4	171	0.649	11	17	1836.000	0.06	**1140**	383.64	$(1/q_{max}, 17, 12)$	**1140**	9.67	(1/10, 14, 8)	**1140**	10.04	(1/5, 10, 5)	**1140**	24.07	**1140**	11
keller5	776	0.752	27	49	28646.982	6.82	≥ 638	o-m	$(1/q_{max}, 49, 19)$	≥ 11236	10800	(1/20, 15, 8)	≥ 11236	10800	(1/10, 7, 4)	≥ 10706	10800	14760	26
keller6	3361	0.818	≥ 59	122	---	---	---	---	$(1/q_{max}, 122, 19)$	≥ 54944	o-m	(1/100, 111, 6)	≥ 29430	o-m	(1/50, 82, 4)	≥ 26496	o-m	136946	50
c125_9	125	0.899	34	44	3176.765	0.14	≥ 119	o-m	$(1/q_{max}, 44, 35)$	**≥ 2766**	o-m	(1/20, 31, 22)	**≥ 2766**	o-m	(1/10, 21, 12)	**≥ 2766**	o-m	**2766**	34
c250_9	250	0.899	44	78	11853.275	1.09	≥ 236	o-m	$(1/q_{max}, 78, 41)$	≥ 7603	o-m	(1/50, 64, 27)	≥ 7603	o-m	(1/20, 41, 13)	≥ 7603	o-m	8123	44
c500_9	500	0.901	≥ 57	144	---	---	---	---	$(1/q_{max}, 144, 45)$	≥ 18669	10800	(1/100, 122, 23)	≥ 18669	10800	(1/50, 89, 15)	≥ 18669	10800	22023	55
c1000_9	1000	0.901	≥ 68	266	---	---	---	---	$(1/q_{max}, 266, 41)$	≥ 35934	10800	(1/100, 172, 14)	≥ 36760	10800	(1/50, 116, 8)	≥ 37578	10800	47098	55
c2000_5	2000	0.500	≥ 16	110	---	---	---	---	$(1/q_{max}, 110, 15)$	**≥ 14344**	10800	(1/50, 102, 7)	**≥ 14334**	10800	(1/20, 46, 2)	≥ 6304	o-m	14344	14
c2000_9	2000	0.900	≥ 80	492	---	---	---	---	$(1/q_{max}, 492, 41)$	≥ 73262	10800	(1/200, 330, 12)	≥ 59299	o-m	(1/100, 226, 3)	≥ 18281	o-m	110318	63
c4000_5	4000	0.500	≥ 18	200	---	---	---	---	$(1/q_{max}, 200, 14)$	**≥ 28642**	10800	(1/100, 150, 5)	≥ 20693	o-m	(1/50, 103, 2)	≥ 8395	o-m	28642	14
MANN_a9	45	0.927	16	18	432.000	0.02	**412**	56.28	$(1/q_{max}, 18, 17)$	**412**	11.64	(1/10, 14, 13)	**412**	11.67	(1/5, 9, 8)	**412**	11.76	**412**	16
MANN_a27	378	0.990	126	137	32606.000	9.73	≥ 374	o-m	$(1/q_{max}, 137, 88)$	≥ 25056	10800	(1/20, 50, 31)	≥ 30316	10800	(1/5, 18, 9)	≥ 30570	10800	**31284**	126
MANN_a45	1035	0.996	345	367	---	---	---	---	$(1/q_{max}, 367, 314)$	≥ 225452	10800	(1/100, 200, 148)	≥ 225452	10800	(1/50, 130, 95)	≥ 232362	10800	**236730**	345
MANN_a81	3321	0.999	≥ 1100	1146	---	---	---	---	$(1/q_{max}, 1146, 13)$	≥ 39524	o-m	(1/200, 480, 0)	---	---	(1/100, 296, 0)	---	---	2437978	1099
d1-RTN	2418	0.0032	10	10	1481.587	0.05	**1273**	0.97	$(1/q_{max}, 10, 9)$	**1273**	0.13	(1/5, 7, 6)	**1273**	0.18	(1/2, 4, 3)	**1273**	0.12	**1273**	8
d3-RTN	4755	0.0024	18	18	4612.926	0.25	**3526**	53.46	$(1/q_{max}, 18, 13)$	**3526**	0.54	(1/10, 14, 9)	**3526**	0.57	(1/5, 9, 5)	**3526**	0.81	**3526**	12
d7-RTN	6511	0.0021	18	18	6945.462	0.50	**5656**	113.05	$(1/q_{max}, 18, 16)$	**5656**	1.60	(1/10, 14, 12)	**5656**	1.69	(1/5, 9, 7)	**5656**	1.97	**5656**	15
d15-RTN	7965	0.0020	18	22	10135.553	0.92	**7772**	557.24	$(1/q_{max}, 22, 17)$	**7772**	2.37	(1/10, 15, 10)	**7772**	2.93	(1/5, 7, 6)	**7772**	11.86	**7772**	16
d30-RTN	10101	0.0018	21	27	16116.109	1.79	**≥ 13099**	o-m	$(1/q_{max}, 27, 22)$	**13099**	3.94	(1/20, 23, 18)	**13099**	4.03	(1/10, 17, 12)	**13099**	4.47	**13099**	21
d66-RTN	13308	0.0017	28	36	28044.977	2.53	≥ 2265	o-m	$(1/q_{max}, 36, 29)$	**22379**	19.18	(1/20, 28, 21)	**22379**	19.86	(1/10, 19, 12)	**22379**	26.16	**22379**	28

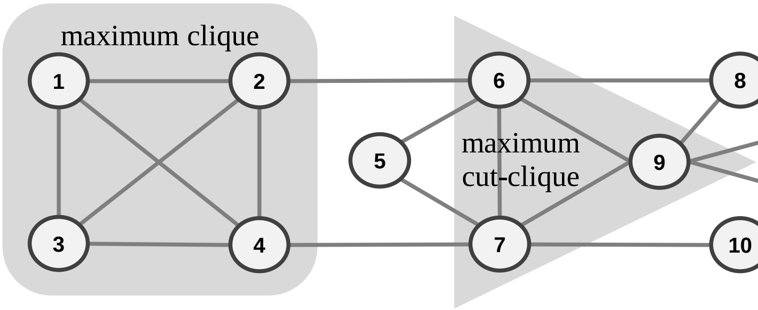
Table 3: Computational results with the model FMCC3 for the MCC problem, with and without the Sequential algorithm.

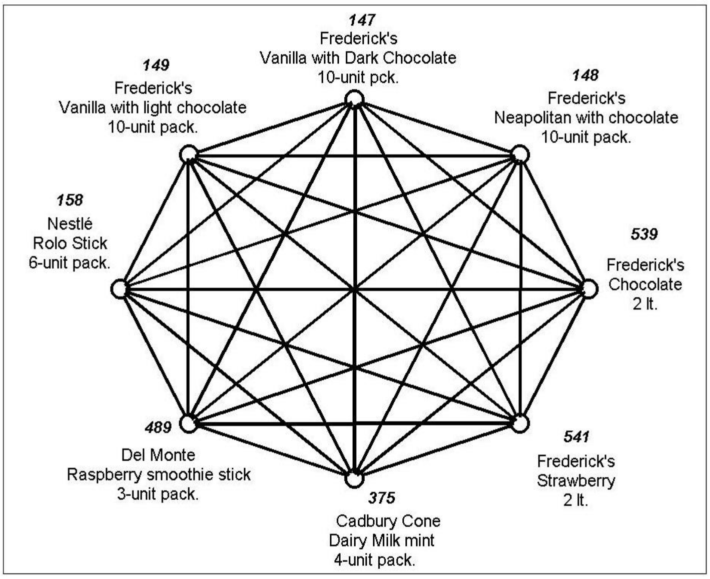
without Proposition 1					**R-ILS**						**D-ILS**					
Instances	n	density	$\omega(G)$	q_{max}	best_sol	$\|C\|$	sol_avg	time_avg	#best	best_sel_avg	best_sol	$\|C\|$	sol_avg	time_avg	#best	best_sel_avg
c-fat200-1	200	0.077	12	12	**81**	9	81	0.1275	100	32.1	**81**	9	81	0.1102	100	28.2
c-fat200-2	200	0.163	24	24	**306**	17	306	0.0738	100	207.0	**306**	17	306	0.1070	100	173.1
c-fat200-5	200	0.426	58	58	**1892**	43	1892	0.0577	100	181.3	**1892**	43	1892	0.1448	100	297.7
c-fat500-1	500	0.036	14	14	**110**	10	110	0.5335	100	30.3	**110**	10	110	0.3823	100	32.5
c-fat500-2	500	0.073	26	26	**380**	19	380	0.3351	100	41.8	**380**	19	380	0.3839	100	40.7
c-fat500-5	500	0.186	64	64	**2304**	48	2304	0.2168	100	792.6	**2304**	48	2304	0.3943	100	551.0
c-fat500-10	500	0.374	126	126	**8930**	94	8930	0.1873	100	742.5	**8930**	94	8930	0.4771	100	683.9
p_hat300-1	300	0.244	8	9	**789**	8	789	0.3255	100	1541.5	**789**	8	789	0.3755	100	309.8
p_hat300-2	300	0.489	25	29	**4637**	25	4637	0.3079	100	353.0	**4637**	25	4637	0.3443	100	98.1
p_hat300-3	300	0.744	36	51	**7740**	36	7740	0.2218	100	591.8	**7740**	36	7740	0.2901	100	410.4
p_hat500-1	500	0.253	9	13	**1621**	9	1621	0.6083	100	4631.5	**1621**	9	1621	0.6988	100	143.6
p_hat500-2	500	0.505	36	46	**11539**	36	11539	0.5291	100	420.9	**11539**	36	11539	0.6362	100	731.4
p_hat500-3	500	0.752	50	78	**18859**	50	18859	0.3975	100	2141.2	**18859**	50	18859	0.5242	100	718.2
p_hat700-1	700	0.249	11	16	**2606**	11	2606	1.1253	100	2015.0	**2606**	11	2606	1.4806	100	1193.2
p_hat700-2	700	0.498	44	60	**20425**	44	20425	0.9902	100	1895.5	**20425**	44	20425	1.3961	100	683.9
p_hat700-3	700	0.748	62	102	**33480**	62	33480	0.8325	100	6383.7	**33480**	62	33480	1.1968	100	1755.2
p_hat1000-1	1000	0.245	≥ 10	20	**3556**	10	3556	1.6477	100	28035.6	**3556**	10	3556	2.1789	100	694.0
p_hat1000-2	1000	0.490	≥ 46	76	**31174**	46	31174	1.5009	100	9373.8	**31174**	46	31174	2.0654	100	471.7
p_hat1000-3	1000	0.744	≥ 68	134	**53259**	68	53259	1.2383	100	1441.8	**53259**	68	53259	1.7756	100	2360.5
p_hat1500-1	1500	0.253	12	28	**6018**	11	5988.10	2.5673	59	44547.6	**6018**	11	6018	3.2721	100	2517.2
p_hat1500-2	1500	0.506	≥ 65	113	**67486**	65	67485.90	2.2242	99	17326.0	**67486**	65	67486	3.0754	100	3672.8
p_hat1500-3	1500	0.754	≥ 94	195	**112873**	94	112872.68	1.8889	92	29083.9	**112873**	94	112873	2.6955	100	7154.2
keller4	171	0.649	11	17	**1140**	11	1140	0.1518	100	70.5	**1140**	11	1140	0.2084	100	59.3
keller5	776	0.752	27	49	**15184**	27	15184	0.8580	100	6104.9	**15184**	27	15116.16	1.1552	84	30780.2
keller6	3361	0.818	≥ 59	122	155060	57	150742.44	3.9152	1	49958.3	149804	55	144873.02	5.0795	5	47112.0
c125_9	125	0.899	34	44	**2766**	34	2766	0.0710	100	763.5	**2766**	34	2766	0.1445	100	129.2
c250_9	250	0.899	44	78	**8123**	44	8123	0.1422	100	8283.9	**8123**	44	8117.57	0.2388	97	2902.9
c500_9	500	0.901	≥ 57	144	**22691**	57	22677.00	0.3399	86	32221.8	**22691**	57	22582.96	0.5032	63	40655.3
c1000_9	1000	0.901	≥ 68	266	**57149**	68	56533.27	1.1472	2	49338.3	**57149**	68	56729.74	1.6825	4	51901.3
c2000_5	2000	0.500	≥ 16	110	16093	16	15768.20	3.9440	1	49294.5	**16106**	16	15951.80	5.0772	1	53890.3
c2000_9	2000	0.900	≥ 80	492	133481	77	130396.18	2.5060	1	54127.2	131843	76	131640.71	3.6068	7	47927.5
c4000_5	4000	0.500	≥ 18	200	36005	18	33825.50	9.2860	1	53924.2	34326	17	34183.54	10.8634	5	48992.4
MANN_a9	45	0.927	16	18	**412**	16	412	0.0298	100	23.1	**412**	16	412	0.0830	100	21.2
MANN_a27	378	0.990	126	137	**31284**	126	31238.40	0.1106	60	29024.5	**31284**	126	30955.24	0.2884	54	28431.0
MANN_a45	1035	0.996	345	367	235090	340	234165.64	0.9317	2	44162.9	236080	343	234738.70	1.6491	5	753.8
MANN_a81	3321	0.999	≥ 1100	1146	2424838	1087	2420749.50	3.5372	5	25549.5	2435808	1097	2434174.50	6.6509	5	1841.7
d1-RTN	2418	0.0032	10	10	**1273**	8	1272.88	0.1543	94	139.2	**1273**	8	1273	0.0515	100	29.7
d3-RTN	4755	0.0024	18	18	**3526**	12	3525.84	0.3882	98	165.8	**3526**	12	3526	0.1139	100	38.0
d7-RTN	6511	0.0021	18	18	**5656**	15	5644.59	0.5351	87	240.9	**5656**	15	5656	0.1535	100	162.2
d15-RTN	7965	0.0020	18	22	**7772**	16	7734.39	0.7217	72	302.4	**7772**	16	7772	0.1932	100	188.5
d30-RTN	10101	0.0018	21	27	**13099**	21	13075.40	0.9527	90	257.1	**13099**	21	13094.48	0.2521	98	151.1
d66-RTN	13308	0.0017	28	36	**22379**	28	22332.61	1.1589	53	361.0	**22379**	28	22368.92	0.3372	91	160.5

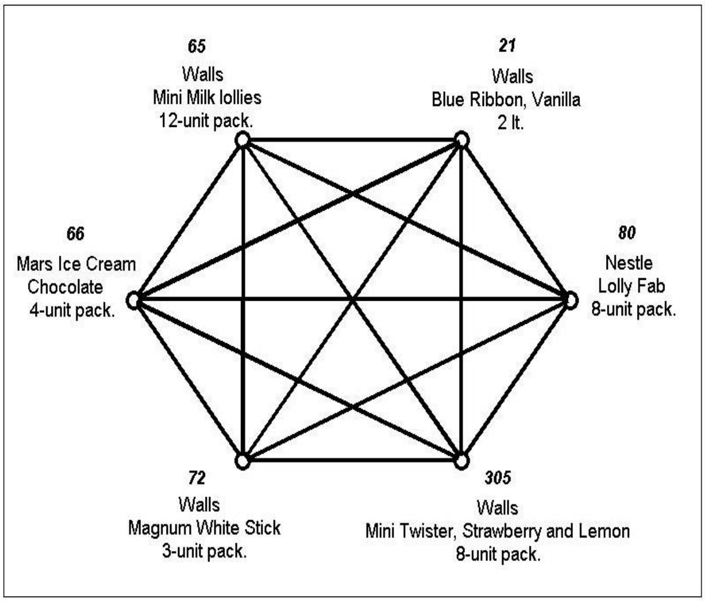
Table 5: Results of the algorithms R-ILS and D-ILS without Proposition 1, for the MCC problem.

with Proposition 1					R-ILS						D-ILS					
Instances	n	density	$\omega(G)$	q_{max}	best_sol	$\|C\|$	sol_avg	time_avg	#best	best_sel_avg	best_sol	$\|C\|$	sol_avg	time_avg	#best	best_sel_avg
c-fat200-1	200	0.077	12	12	**81**	9	81	0.1303	100	34.3	**81**	9	81	0.1105	100	23.4
c-fat200-2	200	0.163	24	24	**306**	17	306	0.0806	100	225.1	**306**	17	306	0.1078	100	146.5
c-fat200-5	200	0.426	58	58	**1892**	43	1892	0.0614	100	193.0	**1892**	43	1892	0.1456	100	211.6
c-fat500-1	500	0.036	14	14	**110**	10	110	0.5407	100	32.8	**110**	10	110	0.3831	100	27.9
c-fat500-2	500	0.073	26	26	**380**	19	380	0.3419	100	45.4	**380**	19	380	0.3847	100	32.7
c-fat500-5	500	0.186	64	64	**2304**	48	2304	0.2243	100	847.1	**2304**	48	2304	0.3955	100	453.2
c-fat500-10	500	0.374	126	126	**8930**	94	8930	0.1929	100	757.9	**8930**	94	8930	0.4782	100	591.0
p_hat300-1	300	0.244	8	9	**789**	8	789	0.3311	100	1572.1	**789**	8	789	0.3765	100	234.3
p_hat300-2	300	0.489	25	29	**4637**	25	4637	0.3138	100	370.6	**4637**	25	4637	0.3451	100	86.3
p_hat300-3	300	0.744	36	51	**7740**	36	7740	0.2270	100	605.5	**7740**	36	7740	0.2908	100	290.0
p_hat500-1	500	0.253	9	13	**1621**	9	1621	0.6175	100	4907.3	**1621**	9	1621	0.7000	100	103.8
p_hat500-2	500	0.505	36	46	**11539**	36	11539	0.5330	100	432.9	**11539**	36	11539	0.6373	100	596.5
p_hat500-3	500	0.752	50	78	**18859**	50	18859	0.4011	100	2259.4	**18859**	50	18859	0.5251	100	585.2
p_hat700-1	700	0.249	11	16	**2606**	11	2606	1.1316	100	2092.4	**2606**	11	2606	1.4819	100	1012.4
p_hat700-2	700	0.498	44	60	**20425**	44	20425	0.9968	100	1926.1	**20425**	44	20425	1.3979	100	490.1
p_hat700-3	700	0.748	62	102	**33480**	62	33480	0.8381	100	6470.7	**33480**	62	33480	1.1981	100	1417.5
p_hat1000-1	1000	0.245	≥ 10	20	**3556**	10	3556	1.6602	100	28192.8	**3556**	10	3556	2.1808	100	524.4
p_hat1000-2	1000	0.490	≥ 46	76	**31174**	46	31174	1.5119	100	9425.5	**31174**	46	31174	2.0671	100	361.7
p_hat1000-3	1000	0.744	≥ 68	134	**53259**	68	53259	1.2405	100	1479.1	**53259**	68	53259	1.7770	100	1960.9
p_hat1500-1	1500	0.253	12	28	**6018**	11	5978.62	2.6407	47	49768.6	**6018**	11	6018	3.2743	100	2067.5
p_hat1500-2	1500	0.506	≥ 65	113	**67486**	65	67486	2.3156	100	19484.0	**67486**	65	67486	3.0778	100	3082.0
p_hat1500-3	1500	0.754	≥ 94	195	**112873**	94	112872.68	1.8897	92	30202.3	**112873**	94	112873	2.6975	100	6232.4
keller4	171	0.649	11	17	**1140**	11	1140	0.1596	100	49.1	**1140**	11	1140	0.2047	100	40.2
keller5	776	0.752	27	49	**15184**	27	15184	0.8849	100	6546.4	**15184**	27	15178.36	1.1555	92	29743.7
keller6	3361	0.818	≥ 59	122	**159130**	59	150411.64	3.8407	1	55274.3	152104	56	145616.06	5.1558	1	48633.1
c125_9	125	0.899	34	44	**2766**	34	2766	0.0734	100	1199.3	**2766**	34	2766	0.1541	100	101.1
c250_9	250	0.899	44	78	**8123**	44	8123	0.1475	100	9201.8	**8123**	44	8123	0.2470	100	2761.5
c500_9	500	0.901	≥ 57	144	**22691**	57	22666.39	0.3418	85	36984.5	**22691**	57	22644.91	0.4971	70	37282.4
c1000_9	1000	0.901	≥ 68	266	**57149**	68	56507.88	1.1918	2	50617.0	**57149**	68	56788.82	1.6683	6	49424.0
c2000_5	2000	0.500	≥ 16	110	16072	16	15751.25	4.2478	1	51356.3	**16106**	16	16009.09	4.6174	1	52111.5
c2000_9	2000	0.900	≥ 80	492	133579	77	130177.36	2.5967	1	48642.4	**133635**	77	132003.97	3.4724	1	48212.7
c4000_5	4000	0.500	≥ 18	200	35861	18	33782.70	9.3830	1	53080.3	**36137**	18	34306.36	10.9307	1	49180.3
MANN_a9	45	0.927	16	18	**412**	16	412	0.0309	100	22.1	**412**	16	412	0.0834	100	19.4
MANN_a27	378	0.990	126	137	**31284**	126	31240.68	0.1133	62	28246.5	**31284**	126	31245.24	0.2905	66	28020.1
MANN_a45	1035	0.996	345	367	235090	340	234189.55	0.9363	1	50853.4	**236406**	344	235715.63	1.6832	1	995.7
MANN_a81	3321	0.999	≥ 1100	1146	2424838	1087	2420749.00	3.5514	4	31213.9	**2436894**	1098	2434828.75	6.7707	3	1824.0
d1-RTN	2418	0.0032	10	10	**1273**	8	1272.98	0.1596	99	157.0	**1273**	8	1273	0.0518	100	24.3
d3-RTN	4755	0.0024	18	18	**3526**	12	3526	0.3927	100	173.8	**3526**	12	3526	0.1145	100	32.5
d7-RTN	6511	0.0021	18	18	**5656**	15	5651.00	0.5525	94	252.6	**5656**	15	5656	0.1548	100	137.2
d15-RTN	7965	0.0020	18	22	**7772**	16	7744.83	0.7292	80	316.5	**7772**	16	7772	0.1948	100	150.3
d30-RTN	10101	0.0018	21	27	**13099**	21	13094.28	0.9660	98	266.2	**13099**	21	13099	0.2535	100	126.8
d66-RTN	13308	0.0017	28	36	**22379**	28	22344.69	1.1674	66	385.3	**22379**	28	22379	0.3397	100	147.1

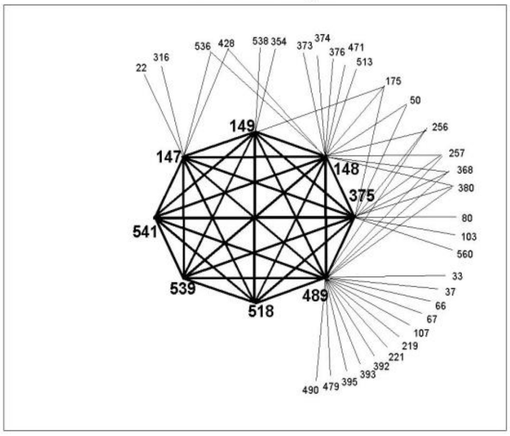
Table 6: Results of the algorithms R-ILS and D-ILS with Proposition 1, for the MCC problem.


maximum clique
1
2
3
4
5
6
7
8
9
10
11
12
13
14
maximum
cut-clique

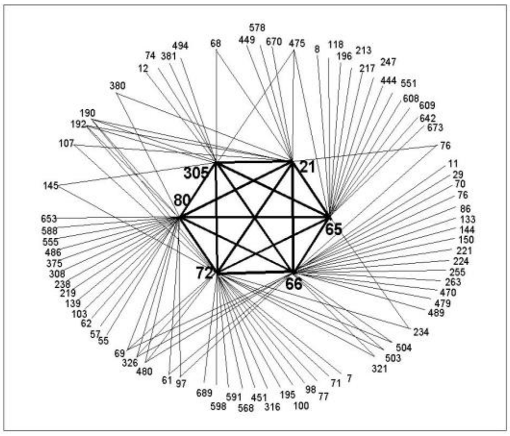

147
Frederick's
Vanilla with Dark Chocolate
10-unit pck.
149
Frederick's
Vanilla with light chocolate
10-unit pack.
148
Frederick's
Neapolitan with chocolate
10-unit pack.
158
Nestlé
Rolo Stick
6-unit pack.
539
Frederick's
Chocolate
2 lt.
489
Del Monte
Raspberry smoothie stick
3-unit pack.
541
Frederick's
Strawberry
2 lt.
375
Cadbury Cone
Dairy Milk mint
4-unit pack.


65
Walls
Mini Milk lollies
12-unit pack.
21
Walls
Blue Ribbon, Vanilla
2 lt.
66
Mars Ice Cream
Chocolate
4-unit pack.
80
Nestle
Lolly Fab
8-unit pack.
72
Walls
Magnum White Stick
3-unit pack.
305
Walls
Mini Twister, Strawberry and Lemon
8-unit pack.

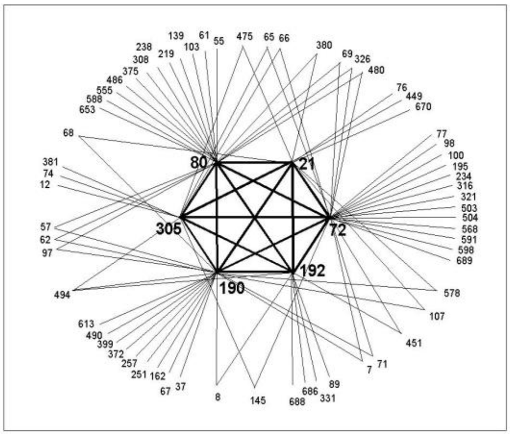
Maximum Clique



Maximum-cut Clique 1



Maximum-cut Clique 2



Instances	n	density	q_{max}	Total number of variable	Test 1		Test 2	
					nº of variables after elimination	% eliminated	nº of variables after elimination	% eliminated
c-fat200-1	200	0.077	12	2400	2400	0.00	1578	34.25
c-fat200-2	200	0.163	24	4800	4800	0.00	3246	32.38
c-fat200-5	200	0.426	58	11600	11600	0.00	8516	26.59
c-fat500-1	500	0.036	14	7000	7000	0.00	4640	33.71
c-fat500-2	500	0.073	26	13000	13000	0.00	9260	28.77
c-fat500-5	500	0.186	64	32000	32000	0.00	23254	27.33
c-fat500-10	500	0.374	126	63000	63000	0.00	46752	25.79
p_hat700-1	700	0.249	16	28000	28000	0.00	27994	0.02
p_hat700-2	700	0.498	60	59500	59500	0.00	59486	0.02
d1-RTN	2418	0.0032	10	24180	13675	43.44	7725	68.05
d3-RTN	4755	0.0024	18	85590	36281	57.61	21128	75.31
d7-RTN	6511	0.0021	18	117198	51664	55.92	31091	73.47
d15-RTN	7965	0.0020	22	175230	69769	60.18	42390	75.81
d30-RTN	10101	0.0018	27	272727	98362	63.93	60291	77.89
d66-RTN	13308	0.0017	36	479088	151839	68.31	94362	80.30

Instances	n	density	$\omega(G)$	q_{max}	FMCC1 LP relaxation opt	FMCC1 LP relaxation time	FMCC1 Branch and bound opt / ≥ best lb	FMCC1 Branch and bound time	FMCC2 LP relaxation opt	FMCC2 LP relaxation time	FMCC2 Branch and bound opt / ≥ best lb	FMCC2 Branch and bound time	Best solution value	Best solution clique size
c-fat200-1	200	0.077	12	12	81.000	0.05	**81**	0.05	81.000	0.02	**81**	0.05	**81**	9
c-fat200-2	200	0.163	24	24	306.000	0.06	**306**	0.09	306.000	0.02	**306**	0.03	**306**	17
c-fat200-5	200	0.426	58	58	1892.000	0.05	**1892**	0.05	1892.000	0.00	**1892**	0.02	**1892**	43
c-fat500-1	500	0.036	14	14	110.000	0.66	**110**	0.76	110.000	0.05	**110**	0.23	**110**	10
c-fat500-2	500	0.073	26	26	380.000	0.64	**380**	0.80	380.000	0.03	**380**	0.23	**380**	19
c-fat500-5	500	0.186	64	64	2304.000	0.75	**2304**	0.83	2304.000	0.03	**2304**	0.16	**2304**	48
c-fat500-10	500	0.374	126	126	8930.000	0.52	**8930**	0.58	8930.000	0.03	**8930**	0.12	**8930**	94
p_hat300-1	300	0.244	8	9	954.000	0.11	**789**	34.73	984.077	0.01	**789**	4.99	**789**	8
p_hat300-2	300	0.489	25	29	5218.500	0.08	**4637**	255.67	5398.621	0.02	**4637**	4050.29	**4637**	25
p_hat300-3	300	0.744	36	51	10035.800	0.01	**≥ 7740**	10800	10379.354	0.02	≥ 7438	o-m	7740	36
p_hat500-1	500	0.253	9	13	2347.500	0.47	**1621**	3227.64	2403.868	0.03	**1621**	251.16	**1621**	9
p_hat500-2	500	0.505	36	46	14204.500	0.30	≥ 11333	o-m	14688.375	0.03	**≥ 11539**	o-m	11539	36
p_hat500-3	500	0.752	50	78	26675.000	0.34	**≥ 18859**	10800	27415.492	0.03	≥ 18305	o-m	18859	50
p_hat700-1	700	0.249	11	16	8794.000	2.01	≥ 2304	o-m	9084.689	0.08	**2606**	2950.82	**2606**	11
p_hat700-2	700	0.498	44	60	34082.500	1.12	≥ 19757	o-m	35913.813	0.05	≥ 19359	o-m	20078	43
p_hat700-3	700	0.748	62	102	61088.333	0.92	≥ 32675	10800	64239.966	0.05	≥ 32228	o-m	33057	61
p_hat1000-1	1000	0.245	≥ 10	20	16065.000	7.99	≥ 3278	o-m	16562.753	0.16	≥ 3385	10800	**3556**	10
p_hat1000-2	1000	0.490	≥ 46	76	62869.500	5.23	≥ 28893	o-m	66070.766	0.11	**≥ 30657**	o-m	30657	45
p_hat1000-3	1000	0.744	≥ 68	134	118907.000	3.09	≥ 40814	10800	123340.573	0.08	**≥ 48894**	o-m	48894	61
p_hat1500-1	1500	0.253	12	28	35502.500	33.21	≥ 4946	o-m	36363.071	0.36	≥ 5923	10800	**6018**	11
p_hat1500-2	1500	0.506	≥ 65	113	141600.500	15.18	≥ 49205	o-m	147468.755	0.28	**≥ 67486**	10800	67486	65
p_hat1500-3	1500	0.754	≥ 94	195	264945.000	10.76	≥ 80610	10800	273833.158	0.20	**≥ 111983**	10800	111983	93
keller4	171	0.649	11	17	1812.000	0.05	**1140**	17.05	1836.000	0.00	**1140**	95.57	**1140**	11
keller5	776	0.752	27	49	28046.000	0.08	**≥ 14760**	10800	28832.179	0.03	≥ 13288	o-m	14760	26
keller6	3361	0.818	≥ 59	122	336256.000	114.68	≥ 105984	o-m	339940.917	0.34	**≥ 136946**	10800	136946	50
c125_9	125	0.899	34	44	3094.500	0.00	**2766**	1.69	3150.306	0.00	**≥ 2766**	o-m	**2766**	34
c250_9	250	0.899	44	78	11680.000	0.02	**≥ 8123**	10800	11801.196	0.02	**≥ 8123**	o-m	8123	44
c500_9	500	0.901	≥ 57	144	44777.500	0.03	**≥ 22023**	10800	45094.729	0.03	≥ 20728	o-m	22023	55
c1000_9	1000	0.901	≥ 68	266	170838.000	0.22	≥ 46055	10800	171616.779	0.14	**≥ 47098**	o-m	47098	55
c2000_5	2000	0.500	≥ 16	110	102128.000	53.74	≥ 12191	o-m	102811.647	0.44	≥ 14180	10800	14344	14
c2000_9	2000	0.900	≥ 80	492	649108.500	9.39	≥ 91790	10800	651142.036	0.91	**≥ 110318**	10800	110318	63
c4000_5	4000	0.500	≥ 18	200	371095.000	497.83	---	---	372789.891	2.31	≥ 26437	o-m	28642	14
MANN_a9	45	0.927	16	18	426.000	0.00	**412**	0.00	426	0.00	**412**	0.25	**412**	16
MANN_a27	378	0.990	126	137	32220.000	0.01	**31284**	1.34	32606.000	0.00	≥ 31054	o-m	**31284**	126
MANN_a45	1035	0.996	345	367	241350.000	0.05	**236730**	106.07	244055.000	0.02	≥ 232362	o-m	**236730**	345
MANN_a81	3321	0.999	≥ 1100	1146	2474658.000	0.25	**≥ 2437978**	10800	2489112.000	0.03	≥ 2417040	o-m	2437978	1099
d1-RTN	2418	0.0032	10	10	1374.000	150.70	**1273**	162.87	1517.848	1.22	**1273**	15.58	**1273**	8
d3-RTN	4755	0.0024	18	18	---	---	---	---	4785.465	5.13	≥ 3460	o-m	**3526**	12
d7-RTN	6511	0.0021	18	18	---	---	---	---	---	---	---	---	**5656**	15
d15-RTN	7965	0.0020	18	22	---	---	---	---	---	---	---	---	**7772**	16
d30-RTN	10101	0.0018	21	27	---	---	---	---	---	---	---	---	**13099**	21
d66-RTN	13308	0.0017	28	36	---	---	---	---	---	---	---	---	**22379**	28

FMCC3					Without the Sequential algorithm (full range)				With the Sequential algorithm										Best solution	
Instances	n	density	$\omega(G)$	q_{max}	LP relaxation		Branch and bound		Partition 1:	Branch and bound		Partition 2:	Branch and bound		Partition 3:	Branch and bound		value	clique size
					opt	time	opt / ≥ best lb	time	(r, k, i)	opt / ≥ best lb	time	(r, k, i)	opt / ≥ best lb	time	(r, k, i)	opt / ≥ best lb	time		
c-fat200-1	200	0.077	12	12	81.000	0.00	**81**	0.02	(1/q_{max}, 12, 10)	**81**	0.02	(1/10, 11, 9)	**81**	0.02	(3/4, 2, 2)	**81**	0.01	**81**	9
c-fat200-2	200	0.163	24	24	306.000	0.03	**306**	0.09	(1/q_{max}, 24, 18)	**306**	0.05	(1/20, 22, 16)	**306**	0.05	(1/2, 5, 3)	**306**	0.07	**306**	17
c-fat200-5	200	0.426	58	58	1892.000	0.22	**1892**	0.61	(1/q_{max}, 58, 44)	**1892**	0.37	(1/50, 54, 40)	**1892**	0.32	(1/10, 23, 12)	**1892**	0.50	**1892**	43
c-fat500-1	500	0.036	14	14	110.000	0.02	**110**	0.08	(1/q_{max}, 14, 11)	**110**	0.03	(1/2, 4, 3)	**110**	0.08	(3/4, 2, 2)	**110**	0.08	**110**	10
c-fat500-2	500	0.073	26	26	380.000	0.11	**380**	0.30	(1/q_{max}, 26, 20)	**380**	0.17	(1/10, 17, 11)	**380**	0.37	(1/5, 11, 6)	**380**	0.30	**380**	19
c-fat500-5	500	0.186	64	64	2304.000	0.67	**2304**	1.90	(1/q_{max}, 64, 49)	**2304**	0.99	(1/20, 39, 22)	**2304**	1.69	(2/3, 3, 3)	**2304**	1.76	**2304**	48
c-fat500-10	500	0.374	126	126	8930.000	3.84	**8930**	9.03	(1/q_{max}, 126, 95)	**8930**	4.45	(1/50, 83, 52)	**8930**	6.03	(1/20, 49, 25)	**8930**	6.44	**8930**	94
p_hat300-1	300	0.244	8	9	967.165	0.03	**789**	32.39	(1/q_{max}, 9, 9)	**789**	2.27	(1/5, 7, 7)	**789**	2.32	(1/2, 4, 4)	**789**	2.41	**789**	8
p_hat300-2	300	0.489	25	29	5390.571	0.36	≥ 229	o-m	(1/q_{max}, 29, 26)	**4637**	128.55	(1/10, 18, 15)	**4637**	128.76	(1/5, 11, 8)	**4637**	147.76	**4637**	25
p_hat300-3	300	0.744	36	51	10388.059	1.14	≥ 267	o-m	(1/q_{max}, 51, 36)	≥ 7587	o-m	(1/20, 33, 18)	≥ 7587	o-m	(1/10, 22, 10)	≥ 7438	o-m	7740	36
p_hat500-1	500	0.253	9	13	2365.519	0.17	**1621**	3796.91	(1/q_{max}, 13, 10)	**1621**	7.22	(1/5, 8, 5)	**1621**	7.40	(1/2, 4, 3)	**1621**	267.07	**1621**	9
p_hat500-2	500	0.505	36	46	14675.660	1.84	≥ 389	o-m	(1/q_{max}, 46, 37)	**≥ 11539**	10800	(1/20, 32, 23)	**≥ 11539**	10800	(1/10, 21, 12)	≥ 11333	10800	11539	36
p_hat500-3	500	0.752	50	78	27424.555	6.04	≥ 452	o-m	(1/q_{max}, 78, 49)	≥ 18305	10800	(1/50, 64, 35)	≥ 18305	10800	(1/20, 41, 17)	≥ 18305	10800	18859	50
p_hat700-1	700	0.249	11	16	8819.206	1.50	≥ 426	o-m	(1/q_{max}, 16, 12)	**2606**	126.14	(1/5, 9, 5)	**2606**	126.31	(1/2, 5, 3)	**2606**	2066.05	**2606**	11
p_hat700-2	700	0.498	44	60	35869.464	9.16	≥ 539	o-m	(1/q_{max}, 60, 44)	**≥ 20078**	10800	(1/20, 36, 20)	**≥ 20078**	10800	(1/10, 24, 11)	≥ 19757	10800	20078	43
p_hat700-3	700	0.748	62	102	---	---	---	---	(1/q_{max}, 102, 61)	**≥ 33057**	10800	(1/50, 75, 35)	**≥ 33057**	10800	(1/20, 45, 15)	≥ 31308	10800	33057	61
p_hat1000-1	1000	0.245	≥ 10	20	16082.792	4.68	≥ 408	o-m	(1/q_{max}, 20, 11)	**3556**	279.83	(1/10, 16, 6)	**3556**	279.83	(1/5, 10, 4)	**≥ 3556**	10800	**3556**	10
p_hat1000-2	1000	0.490	≥ 46	76	---	---	---	---	(1/q_{max}, 76, 46)	**≥ 30657**	10800	(1/50, 63, 33)	**≥ 30657**	10800	(1/20, 40, 15)	**≥ 30657**	10800	30657	45
p_hat1000-3	1000	0.744	≥ 68	134	---	---	---	---	(1/q_{max}, 134, 62)	**≥ 48894**	10800	(1/100, 117, 45)	**≥ 48894**	10800	(1/50, 86, 25)	≥ 48184	10800	48894	61
p_hat1500-1	1500	0.253	12	28	---	---	---	---	(1/q_{max}, 28, 12)	**6018**	7014.07	(1/20, 24, 8)	**6018**	7015.08	(1/10, 17, 5)	**≥ 6018**	10800	**6018**	11
p_hat1500-2	1500	0.506	≥ 65	113	---	---	---	---	(1/q_{max}, 113, 59)	≥ 62108	10800	(1/50, 79, 27)	≥ 60206	10800	(1/20, 48, 13)	≥ 62108	10800	67486	65
p_hat1500-3	1500	0.754	≥ 94	195	---	---	---	---	(1/q_{max}, 195, 73)	≥ 90095	10800	(1/100, 147, 36)	≥ 88955	10800	(1/50, 102, 20)	≥ 88955	10800	111983	93
keller4	171	0.649	11	17	1836.000	0.06	**1140**	383.64	(1/q_{max}, 17, 12)	**1140**	9.67	(1/10, 14, 8)	**1140**	10.04	(1/5, 10, 5)	**1140**	24.07	**1140**	11
keller5	776	0.752	27	49	28646.982	6.82	≥ 638	o-m	(1/q_{max}, 49, 19)	≥ 11236	10800	(1/20, 15, 8)	≥ 11236	10800	(1/10, 7, 4)	≥ 10706	10800	14760	26
keller6	3361	0.818	≥ 59	122	---	---	---	---	(1/q_{max}, 122, 19)	≥ 54944	o-m	(1/100, 111, 6)	≥ 29430	o-m	(1/50, 82, 4)	≥ 26496	o-m	136946	50
c125_9	125	0.899	34	44	3176.765	0.14	≥ 119	o-m	(1/q_{max}, 44, 35)	**≥ 2766**	o-m	(1/20, 31, 22)	**≥ 2766**	o-m	(1/10, 21, 12)	**≥ 2766**	o-m	**2766**	34
c250_9	250	0.899	44	78	11853.275	1.09	≥ 236	o-m	(1/q_{max}, 78, 41)	≥ 7603	o-m	(1/50, 64, 27)	≥ 7603	o-m	(1/20, 41, 13)	≥ 7603	o-m	8123	44
c500_9	500	0.901	≥ 57	144	---	---	---	---	(1/q_{max}, 144, 45)	≥ 18669	10800	(1/100, 122, 23)	≥ 18669	10800	(1/50, 89, 15)	≥ 18669	10800	22023	55
c1000_9	1000	0.901	≥ 68	266	---	---	---	---	(1/q_{max}, 266, 41)	≥ 35934	10800	(1/100, 172, 14)	≥ 36760	10800	(1/50, 116, 8)	≥ 37578	10800	47098	55
c2000_5	2000	0.500	≥ 16	110	---	---	---	---	(1/q_{max}, 110, 15)	**≥ 14344**	10800	(1/50, 102, 7)	**≥ 14334**	10800	(1/20, 46, 2)	≥ 6304	o-m	14344	14
c2000_9	2000	0.900	≥ 80	492	---	---	---	---	(1/q_{max}, 492, 41)	≥ 73262	10800	(1/200, 330, 12)	≥ 59299	o-m	(1/100, 226, 3)	≥ 18281	o-m	110318	63
c4000_5	4000	0.500	≥ 18	200	---	---	---	---	(1/q_{max}, 200, 14)	**≥ 28642**	10800	(1/100, 150, 5)	≥ 20693	o-m	(1/50, 103, 2)	≥ 8395	o-m	28642	14
MANN_a9	45	0.927	16	18	432.000	0.02	**412**	56.28	(1/q_{max}, 18, 17)	**412**	11.64	(1/10, 14, 13)	**412**	11.67	(1/5, 9, 8)	**412**	11.76	**412**	16
MANN_a27	378	0.990	126	137	32606.000	9.73	≥ 374	o-m	(1/q_{max}, 137, 88)	≥ 25056	10800	(1/20, 50, 31)	≥ 30316	10800	(1/5, 18, 9)	≥ 30570	10800	**31284**	126
MANN_a45	1035	0.996	345	367	---	---	---	---	(1/q_{max}, 367, 314)	≥ 225452	10800	(1/100, 200, 148)	≥ 225452	10800	(1/50, 130, 95)	≥ 232362	10800	**236730**	345
MANN_a81	3321	0.999	≥ 1100	1146	---	---	---	---	(1/q_{max}, 1146, 13)	≥ 39524	o-m	(1/200, 480, 0)	---	---	(1/100, 296, 0)	---	---	2437978	1099
d1-RTN	2418	0.0032	10	10	1481.587	0.05	**1273**	0.97	(1/q_{max}, 10, 9)	**1273**	0.13	(1/5, 7, 6)	**1273**	0.18	(1/2, 4, 3)	**1273**	0.12	**1273**	8
d3-RTN	4755	0.0024	18	18	4612.926	0.25	**3526**	53.46	(1/q_{max}, 18, 13)	**3526**	0.54	(1/10, 14, 9)	**3526**	0.57	(1/5, 9, 5)	**3526**	0.81	**3526**	12
d7-RTN	6511	0.0021	18	18	6945.462	0.50	**5656**	113.05	(1/q_{max}, 18, 16)	**5656**	1.60	(1/10, 14, 12)	**5656**	1.69	(1/5, 9, 7)	**5656**	1.97	**5656**	15
d15-RTN	7965	0.0020	18	22	10135.553	0.92	**7772**	557.24	(1/q_{max}, 22, 17)	**7772**	2.37	(1/10, 15, 10)	**7772**	2.93	(1/5, 7, 6)	**7772**	11.86	**7772**	16
d30-RTN	10101	0.0018	21	27	16116.109	1.79	**≥ 13099**	o-m	(1/q_{max}, 27, 22)	**13099**	3.94	(1/20, 23, 18)	**13099**	4.03	(1/10, 17, 12)	**13099**	4.47	**13099**	21
d66-RTN	13308	0.0017	28	36	28044.977	2.53	≥ 2265	o-m	(1/q_{max}, 36, 29)	**22379**	19.18	(1/20, 28, 21)	**22379**	19.86	(1/10, 19, 12)	**22379**	26.16	**22379**	28

without Proposition 1					**R-ILS**						**D-ILS**					
Instances	n	density	$\omega(G)$	q_{max}	best_sol	$\|C\|$	sol_avg	time_avg	#best	best_sel_avg	best_sol	$\|C\|$	sol_avg	time_avg	#best	best_sel_avg
c-fat200-1	200	0.077	12	12	**81**	9	81	0.1275	100	32.1	**81**	9	81	0.1102	100	28.2
c-fat200-2	200	0.163	24	24	**306**	17	306	0.0738	100	207.0	**306**	17	306	0.1070	100	173.1
c-fat200-5	200	0.426	58	58	**1892**	43	1892	0.0577	100	181.3	**1892**	43	1892	0.1448	100	297.7
c-fat500-1	500	0.036	14	14	**110**	10	110	0.5335	100	30.3	**110**	10	110	0.3823	100	32.5
c-fat500-2	500	0.073	26	26	**380**	19	380	0.3351	100	41.8	**380**	19	380	0.3839	100	40.7
c-fat500-5	500	0.186	64	64	**2304**	48	2304	0.2168	100	792.6	**2304**	48	2304	0.3943	100	551.0
c-fat500-10	500	0.374	126	126	**8930**	94	8930	0.1873	100	742.5	**8930**	94	8930	0.4771	100	683.9
p_hat300-1	300	0.244	8	9	**789**	8	789	0.3255	100	1541.5	**789**	8	789	0.3755	100	309.8
p_hat300-2	300	0.489	25	29	**4637**	25	4637	0.3079	100	353.0	**4637**	25	4637	0.3443	100	98.1
p_hat300-3	300	0.744	36	51	**7740**	36	7740	0.2218	100	591.8	**7740**	36	7740	0.2901	100	410.4
p_hat500-1	500	0.253	9	13	**1621**	9	1621	0.6083	100	4631.5	**1621**	9	1621	0.6988	100	143.6
p_hat500-2	500	0.505	36	46	**11539**	36	11539	0.5291	100	420.9	**11539**	36	11539	0.6362	100	731.4
p_hat500-3	500	0.752	50	78	**18859**	50	18859	0.3975	100	2141.2	**18859**	50	18859	0.5242	100	718.2
p_hat700-1	700	0.249	11	16	**2606**	11	2606	1.1253	100	2015.0	**2606**	11	2606	1.4806	100	1193.2
p_hat700-2	700	0.498	44	60	**20425**	44	20425	0.9902	100	1895.5	**20425**	44	20425	1.3961	100	683.9
p_hat700-3	700	0.748	62	102	**33480**	62	33480	0.8325	100	6383.7	**33480**	62	33480	1.1968	100	1755.2
p_hat1000-1	1000	0.245	≥ 10	20	**3556**	10	3556	1.6477	100	28035.6	**3556**	10	3556	2.1789	100	694.0
p_hat1000-2	1000	0.490	≥ 46	76	**31174**	46	31174	1.5009	100	9373.8	**31174**	46	31174	2.0654	100	471.7
p_hat1000-3	1000	0.744	≥ 68	134	**53259**	68	53259	1.2383	100	1441.8	**53259**	68	53259	1.7756	100	2360.5
p_hat1500-1	1500	0.253	12	28	**6018**	11	5988.10	2.5673	59	44547.6	**6018**	11	6018	3.2721	100	2517.2
p_hat1500-2	1500	0.506	≥ 65	113	**67486**	65	67485.90	2.2242	99	17326.0	**67486**	65	67486	3.0754	100	3672.8
p_hat1500-3	1500	0.754	≥ 94	195	**112873**	94	112872.68	1.8889	92	29083.9	**112873**	94	112873	2.6955	100	7154.2
keller4	171	0.649	11	17	**1140**	11	1140	0.1518	100	70.5	**1140**	11	1140	0.2084	100	59.3
keller5	776	0.752	27	49	**15184**	27	15184	0.8580	100	6104.9	**15184**	27	15116.16	1.1552	84	30780.2
keller6	3361	0.818	≥ 59	122	155060	57	150742.44	3.9152	1	49958.3	149804	55	144873.02	5.0795	5	47112.0
c125_9	125	0.899	34	44	**2766**	34	2766	0.0710	100	763.5	**2766**	34	2766	0.1445	100	129.2
c250_9	250	0.899	44	78	**8123**	44	8123	0.1422	100	8283.9	**8123**	44	8117.57	0.2388	97	2902.9
c500_9	500	0.901	≥ 57	144	**22691**	57	22677.00	0.3399	86	32221.8	**22691**	57	22582.96	0.5032	63	40655.3
c1000_9	1000	0.901	≥ 68	266	**57149**	68	56533.27	1.1472	2	49338.3	**57149**	68	56729.74	1.6825	4	51901.3
c2000_5	2000	0.500	≥ 16	110	16093	16	15768.20	3.9440	1	49294.5	**16106**	16	15951.80	5.0772	1	53890.3
c2000_9	2000	0.900	≥ 80	492	133481	77	130396.18	2.5060	1	54127.2	131843	76	131640.71	3.6068	7	47927.5
c4000_5	4000	0.500	≥ 18	200	36005	18	33825.50	9.2860	1	53924.2	34326	17	34183.54	10.8634	5	48992.4
MANN_a9	45	0.927	16	18	**412**	16	412	0.0298	100	23.1	**412**	16	412	0.0830	100	21.2
MANN_a27	378	0.990	126	137	**31284**	126	31238.40	0.1106	60	29024.5	**31284**	126	30955.24	0.2884	54	28431.0
MANN_a45	1035	0.996	345	367	235090	340	234165.64	0.9317	2	44162.9	236080	343	234738.70	1.6491	5	753.8
MANN_a81	3321	0.999	≥ 1100	1146	2424838	1087	2420749.50	3.5372	5	25549.5	2435808	1097	2434174.50	6.6509	5	1841.7
d1-RTN	2418	0.0032	10	10	**1273**	8	1272.88	0.1543	94	139.2	**1273**	8	1273	0.0515	100	29.7
d3-RTN	4755	0.0024	18	18	**3526**	12	3525.84	0.3882	98	165.8	**3526**	12	3526	0.1139	100	38.0
d7-RTN	6511	0.0021	18	18	**5656**	15	5644.59	0.5351	87	240.9	**5656**	15	5656	0.1535	100	162.2
d15-RTN	7965	0.0020	18	22	**7772**	16	7734.39	0.7217	72	302.4	**7772**	16	7772	0.1932	100	188.5
d30-RTN	10101	0.0018	21	27	**13099**	21	13075.40	0.9527	90	257.1	**13099**	21	13094.48	0.2521	98	151.1
d66-RTN	13308	0.0017	28	36	**22379**	28	22332.61	1.1589	53	361.0	**22379**	28	22368.92	0.3372	91	160.5

with Proposition 1					**R-ILS**						**D-ILS**					
Instances	n	density	$\omega(G)$	q_{max}	best_sol	$\|C\|$	sol_avg	time_avg	#best	best_sel_avg	best_sol	$\|C\|$	sol_avg	time_avg	#best	best_sel_avg
c-fat200-1	200	0.077	12	12	**81**	9	81	0.1303	100	34.3	**81**	9	81	0.1105	100	23.4
c-fat200-2	200	0.163	24	24	**306**	17	306	0.0806	100	225.1	**306**	17	306	0.1078	100	146.5
c-fat200-5	200	0.426	58	58	**1892**	43	1892	0.0614	100	193.0	**1892**	43	1892	0.1456	100	211.6
c-fat500-1	500	0.036	14	14	**110**	10	110	0.5407	100	32.8	**110**	10	110	0.3831	100	27.9
c-fat500-2	500	0.073	26	26	**380**	19	380	0.3419	100	45.4	**380**	19	380	0.3847	100	32.7
c-fat500-5	500	0.186	64	64	**2304**	48	2304	0.2243	100	847.1	**2304**	48	2304	0.3955	100	453.2
c-fat500-10	500	0.374	126	126	**8930**	94	8930	0.1929	100	757.9	**8930**	94	8930	0.4782	100	591.0
p_hat300-1	300	0.244	8	9	**789**	8	789	0.3311	100	1572.1	**789**	8	789	0.3765	100	234.3
p_hat300-2	300	0.489	25	29	**4637**	25	4637	0.3138	100	370.6	**4637**	25	4637	0.3451	100	86.3
p_hat300-3	300	0.744	36	51	**7740**	36	7740	0.2270	100	605.5	**7740**	36	7740	0.2908	100	290.0
p_hat500-1	500	0.253	9	13	**1621**	9	1621	0.6175	100	4907.3	**1621**	9	1621	0.7000	100	103.8
p_hat500-2	500	0.505	36	46	**11539**	36	11539	0.5330	100	432.9	**11539**	36	11539	0.6373	100	596.5
p_hat500-3	500	0.752	50	78	**18859**	50	18859	0.4011	100	2259.4	**18859**	50	18859	0.5251	100	585.2
p_hat700-1	700	0.249	11	16	**2606**	11	2606	1.1316	100	2092.4	**2606**	11	2606	1.4819	100	1012.4
p_hat700-2	700	0.498	44	60	**20425**	44	20425	0.9968	100	1926.1	**20425**	44	20425	1.3979	100	490.1
p_hat700-3	700	0.748	62	102	**33480**	62	33480	0.8381	100	6470.7	**33480**	62	33480	1.1981	100	1417.5
p_hat1000-1	1000	0.245	≥ 10	20	**3556**	10	3556	1.6602	100	28192.8	**3556**	10	3556	2.1808	100	524.4
p_hat1000-2	1000	0.490	≥ 46	76	**31174**	46	31174	1.5119	100	9425.5	**31174**	46	31174	2.0671	100	361.7
p_hat1000-3	1000	0.744	≥ 68	134	**53259**	68	53259	1.2405	100	1479.1	**53259**	68	53259	1.7770	100	1960.9
p_hat1500-1	1500	0.253	12	28	**6018**	11	5978.62	2.6407	47	49768.6	**6018**	11	6018	3.2743	100	2067.5
p_hat1500-2	1500	0.506	≥ 65	113	**67486**	65	67486	2.3156	100	19484.0	**67486**	65	67486	3.0778	100	3082.0
p_hat1500-3	1500	0.754	≥ 94	195	**112873**	94	112872.68	1.8897	92	30202.3	**112873**	94	112873	2.6975	100	6232.4
keller4	171	0.649	11	17	**1140**	11	1140	0.1596	100	49.1	**1140**	11	1140	0.2047	100	40.2
keller5	776	0.752	27	49	**15184**	27	15184	0.8849	100	6546.4	**15184**	27	15178.36	1.1555	92	29743.7
keller6	3361	0.818	≥ 59	122	**159130**	59	150411.64	3.8407	1	55274.3	152104	56	145616.06	5.1558	1	48633.1
c125_9	125	0.899	34	44	**2766**	34	2766	0.0734	100	1199.3	**2766**	34	2766	0.1541	100	101.1
c250_9	250	0.899	44	78	**8123**	44	8123	0.1475	100	9201.8	**8123**	44	8123	0.2470	100	2761.5
c500_9	500	0.901	≥ 57	144	**22691**	57	22666.39	0.3418	85	36984.5	**22691**	57	22644.91	0.4971	70	37282.4
c1000_9	1000	0.901	≥ 68	266	**57149**	68	56507.88	1.1918	2	50617.0	**57149**	68	56788.82	1.6683	6	49424.0
c2000_5	2000	0.500	≥ 16	110	16072	16	15751.25	4.2478	1	51356.3	**16106**	16	16009.09	4.6174	1	52111.5
c2000_9	2000	0.900	≥ 80	492	133579	77	130177.36	2.5967	1	48642.4	**133635**	77	132003.97	3.4724	1	48212.7
c4000_5	4000	0.500	≥ 18	200	35861	18	33782.70	9.3830	1	53080.3	**36137**	18	34306.36	10.9307	1	49180.3
MANN_a9	45	0.927	16	18	**412**	16	412	0.0309	100	22.1	**412**	16	412	0.0834	100	19.4
MANN_a27	378	0.990	126	137	**31284**	126	31240.68	0.1133	62	28246.5	**31284**	126	31245.24	0.2905	66	28020.1
MANN_a45	1035	0.996	345	367	235090	340	234189.55	0.9363	1	50853.4	**236406**	344	235715.63	1.6832	1	995.7
MANN_a81	3321	0.999	≥ 1100	1146	2424838	1087	2420749.00	3.5514	4	31213.9	**2436894**	1098	2434828.75	6.7707	3	1824.0
d1-RTN	2418	0.0032	10	10	**1273**	8	1272.98	0.1596	99	157.0	**1273**	8	1273	0.0518	100	24.3
d3-RTN	4755	0.0024	18	18	**3526**	12	3526	0.3927	100	173.8	**3526**	12	3526	0.1145	100	32.5
d7-RTN	6511	0.0021	18	18	**5656**	15	5651.00	0.5525	94	252.6	**5656**	15	5656	0.1548	100	137.2
d15-RTN	7965	0.0020	18	22	**7772**	16	7744.83	0.7292	80	316.5	**7772**	16	7772	0.1948	100	150.3
d30-RTN	10101	0.0018	21	27	**13099**	21	13094.28	0.9660	98	266.2	**13099**	21	13099	0.2535	100	126.8
d66-RTN	13308	0.0017	28	36	**22379**	28	22344.69	1.1674	66	385.3	**22379**	28	22379	0.3397	100	147.1