An annotated bibliography of combinatorial optimization problems with fixed cardinality constraints

Maurizio Bruglieri\textsuperscript{a}, Matthias Ehrgott\textsuperscript{b,1}, Horst W. Hamacher\textsuperscript{c,2}, Francesco Maffioli\textsuperscript{a}

\textsuperscript{a}Dipartimento di Elettronica e Informazione, Politecnico di Milano, 20133 Milano, Italy
\textsuperscript{b}Department of Engineering Science, The University of Auckland, Auckland, New Zealand
\textsuperscript{c}Fachbereich Mathematik, Universität Kaiserslautern, 67663 Kaiserslautern, Germany

Received 1 October 2003; received in revised form 7 April 2004; accepted 6 May 2005
Available online 25 January 2006

Abstract

In this paper, we consider combinatorial optimization problems with additional cardinality constraints. In $k$-cardinality combinatorial optimization problems, a cardinality constraint requires feasible solutions to contain exactly $k$ elements of a finite set $E$. Problems of this type have applications in many areas, e.g. in the mining and oil industry, telecommunications, circuit layout, and location planning. We formally define the problem, mention some examples and summarize general results. We provide an annotated bibliography of combinatorial optimization problems of which versions with cardinality constraint have been considered in the literature.

© 2005 Elsevier B.V. All rights reserved.

MSC: 90C27

Keywords: Combinatorial optimization; Cardinality constraint; Annotated bibliography

1. Introduction

A combinatorial optimization problem is given by a finite set $E$ with cardinality $|E| = m$, the set of feasible solutions $\mathcal{X}$, i.e. a family $\mathcal{X} \subseteq 2^E$ of subsets of $E$ (generally not given explicitly), and an objective function $f : \mathcal{X} \rightarrow \mathbb{N}$ assigning to each feasible solution $x \in \mathcal{X}$ its non-negative integer objective value $f(x)$. The mathematical program

$$\min_{x \in \mathcal{X}} f(x),$$

...
is the general form of a combinatorial optimization problem (COP). If the context is clear, we often identify $x$ with its incidence vector $\tilde{x} \in \{0, 1\}^m$ defined by

$$\tilde{x}_e = \begin{cases} 1 & \text{if } e \in x, \\ 0 & \text{otherwise}. \end{cases}$$

In this paper additional cardinality constraints are considered. If $k$ is any natural number with $1 \leq k \leq m$ we define $\mathcal{X}_k := \{x \in \mathcal{X} : |x| = k\}$, the family of feasible solutions with cardinality $k$. Here $|x| = \sum_{e \in E} \tilde{x}_e$ denotes the cardinality of $x$. The $k$-cardinality combinatorial optimization problem is given as

$$\min_{x \in \mathcal{X}_k} f(x). \quad (k\text{CardCOP})$$

While many COPs have a natural constraint on the cardinality of the feasible solutions (e.g., the TSP, the matroid base problem, the assignment problem), such a constraint can be imposed in addition to the problem constraints for all other COPs. In this paper we are concerned with the latter type of problems.

Combinatorial optimization problems with such a cardinality constraint are relevant in many applications. In the mining and oil industries it can be the case that in a mine or oilfield, subdivided into unit blocks, only a certain subfield can be explored or mined. This requirement is modelled as the problem of selecting the desired number $k$ of blocks subject to some constraints, usually to guarantee connectedness of the chosen subfield so that equipment can be moved throughout the mined area. This problem leads to the formulation of the $k$-cardinality tree problem; see Section 2. This model also has applications in facility layout, telecommunication and routing problems. When a salesperson has to visit $k$ out of $n$ customers, we encounter $k$-cardinality TSP problems; see Section 3. The problem of partitioning the node set of a graph into $k$ subsets of almost equal size has applications in circuit layout; see Section 5. Many location problems involve the selection of $k$ sites for the placement of facilities such as shops, machines, etc. (Section 7).

In this paper we provide an annotated bibliography of the research on this type of combinatorial optimization problems. The main interest will be the description of the effects of the cardinality constraint. The questions covered include the computational complexity of problems with added cardinality constraint and methods for their solution. In this introductory section we provide some general statements, whereas in the subsequent ones we give an overview on the literature problem by problem.

For now, we consider three types of problems. In the existence problem we want to decide whether the feasible set $\mathcal{X}_k$ is non-empty. Therefore for all $E' \subseteq E$ we define $E_x(E')$ as the problem of establishing whether

$$\{x \in \mathcal{X} : |x| = k \text{ and } x \subseteq E'\} \neq \emptyset.$$  

In the sum problem and the minmax (or bottleneck) problem we want to find minimizers of $k\text{CardCOP}$, where the objective function is defined using a weight function $w : E \to \mathbb{N}$ as

$$f(x) = \sum_{e \in x} w(e) = \sum_{e \in E} w(e)\tilde{x}_e$$

and

$$f(x) = \max_{e \in x} w(e) = \max_{e \in E} w(e)\tilde{x}_e,$$

respectively. As is the case with the solution of bottleneck COPs in general, $k\text{CardCOP}$s with bottleneck objective can be solved efficiently if and only if the same is true for existence problems, using a threshold method (see [75,72]).

**Remark 1.** Given an instance of $k\text{CardCOP}$ with finite set $E$ and weight function $w : E \to \mathbb{N}$, then the bottleneck problem is solvable in polynomial-time if and only if the existence problem $E_x(E')$ is polynomially solvable for any subset $E' \subseteq E$.

An example of a bottleneck $k\text{CardCOP}$ which can be solved in polynomial-time using Remark 1 is the $k\text{Card}$ tree problem. The existence problem is the check for a subtree in $G' = (V, E')$ with cardinality $k$. This can easily be done in polynomial-time.
On the other hand, few examples of polynomially solvable \( k \text{CardCOP} \) with sum objective are known. One such class are problems where the COP without cardinality constraints is solved by a primal-dual algorithm (or augmenting structure algorithm). Corresponding \( k \text{CardCOP} \) include the cardinality constrained versions of the following problems:

- assignment,
- non-bipartite matching,
- acyclic subgraph,
- matroid base and independent set,
- matroid intersection.

Conversely, every algorithm to solve \( k \text{CardCOP} \) can obviously be used to solve COP without cardinality constraints, by

\[
\min_{x \in \mathcal{X}} f(x) = \min_{k=1, \ldots, |E|} \{ \min_{x \in \mathcal{X}_k} f(x) : x \in \mathcal{X}_k \}.
\]

In the same way, \( k \text{CardCOP} \) can be used to solve COP with cardinality constraints \(|x| \leq k \) (\( \leq k \text{CardCOP} \)). Thus, \( k \)-cardinality COPs are always at least as hard as their counterparts without fixed cardinality and also \( \leq k \)-cardinality COPs. However, we have to note that any \( k \)-cardinality combinatorial optimization problem can be solved in polynomial time, if \( k \) is fixed, i.e., not part of the problem input. A simple brute force enumeration of the at most \( O(n^k) \) feasible sets with cardinality \( k \) will work. In this sense, \( k \text{CardCOP} \) is a special case of fixed parameter optimization as discussed in [53] where the main issue is whether there is an algorithm with running time \( f(k)n^z \) where \( f \) is an arbitrary function depending only on \( k, n \) is the dimension of the instance and \( z \) is a constant. In this theory, problem parameters are fixed and the resulting impact on tractability and complexity is studied. All \( \mathbb{NP} \)-hardness results mentioned in this paper assume \( k \) to be part of the input.

Finally, we remark that bottleneck problems and also lexicographic bottleneck problems (i.e., the problem of minimizing also the second largest weight, the third largest weight, and so on) are not harder than the corresponding problem with sum objective as shown in [34]. This is of course also true for \( k \text{CardCOPs} \).

In accordance with these general observations, most research on \( k \text{CardCOPs} \) has focused on sum problems. As we shall see during the course of the annotated bibliography, the addition of a fixed cardinality constraint makes otherwise polynomially solvable COPs \( \mathbb{NP} \)-hard. This is the case for, e.g., the minimum spanning tree, minimal weight cut and shortest path problem.

Polyhedral theory is an important tool in combinatorial optimization. Various authors have investigated the particular polyhedral structure of some \( k \text{CardCOPs} \), see, e.g. Fischetti et al. [67] for the \( k \)Card tree problem, Gamble and Pulleyblank [73] for the forest cover problem, White and Gillerson [131] for the node \( k \)-cover and Ward et al. [128] for \( k \)-median on tree networks. To the best of our knowledge the only reference on the polyhedral structure of the general \( k \text{CardCOP} \) is that of Aghezzaf and Wolsey [3]. If

\[
P_k := \text{conv}(T) \cap \left\{ (x, y) \in \mathbb{R}^n_+ \times [0, 1]^n : \sum_{j=1}^n y_j = k \right\},
\]

where \( T \subset \mathbb{R}^p_+ \times \{0, 1\}^n \) represents a general COP, they consider the linear program

\[ F(k) := \min \{ cx + fy : (x, y) \in P_k \} \]

and the integer program

\[ G(k) := \min \{ cx + fy : (x, y) \in P_k, y \in \{0, 1\}^n \}. \]

COP has the integral property with respect to \((c, f)\) if \( F(k) = G(k) \), i.e., \( F(k) \) has an optimal solution in \( T \) for all \( k \). Aghezzaf and Wolsey [3] succeeded to prove the following theorem.
Theorem 1. Given an objective function \((c, f)\), the following are equivalent.

1. The integral property with respect to \((c, f)\) holds.
2. \(G(k+1) - G(k)\) is nondecreasing for \(k = 1, \ldots, n - 1\).
3. Whenever for some value of \(\mu\) the linear program \(\min\{cx + f y - \mu \sum j y_j : (x, y) \in \text{conv}(T)\}\) has two integral optimal solutions \((x', y')\) and \((x'', y'')\) with \(\sum y'_j = k'\), \(\sum y''_j = k''\) and \(k'' - k' > 2\), there exists an alternative integral optimal solution \((x^*, y^*)\) with \(\sum y^*_j = k\) and \(k' < k < k''\).

The authors show in the same paper that the uncapacitated lot-sizing polyhedron with a cardinality constraint has the integral property when the production costs are non-increasing over time since it satisfies the third condition of Theorem 1. Bruglieri et al. [32] have observed that for the \(k\)Card cut problem (a \(k\)Card cut is a cut with exactly \(k\) edges) property 3 is trivially satisfied for trees, but does not hold for planar graphs.

In the following sections we present a review of the existing literature on fixed cardinality optimization problems. We include only references where an explicit fixed cardinality constraint on the feasible solutions is given. The huge amount of literature that deals with closely related problems, such as additional linear constraints (e.g., the shortest path problem with a knapsack-type constraint), or to find solutions of cardinality approximately \(k\) (e.g., [105]), or other types of fixed cardinalities (e.g., solutions containing a fixed number of elements of a certain set) is not considered. We also left out COPs where the cardinality is bounded by some \(k \in \mathbb{N} (\leq k\text{-CardCOP})\). The description below is organized chronologically according to the evolution of the results in research reports. We cite, however, the corresponding articles when published in journals for ease of reference.

2. The \(k\)-cardinality tree problem

All the relevant aspects of the \(k\)-cardinality tree problem are subject of research, such as complexity, approximation algorithms, integer programming approaches and heuristics. This fact may be due to the wide variety of applications. We found references to such diverse fields as oil-field leasing (Hamacher and Joernsten [84]), facility layout (Foulds [107]), open-pit mining (Philpott and Wormald [117]), and telecommunications (Garg and Hochbaum [77]). But it is also applied within combinatorial optimization itself, e.g., as a subproblem in matrix decomposition, see [28,29].

Given a graph \(G = (V, E)\) with a weight function on the edges or the vertices, the objective is to find a subtree of \(G\) containing exactly \(k\) edges (or, equivalently, \(k + 1\) vertices) such that the sum of the weights is minimal. A majority of the existing articles have considered the edge-weighted case.

Several authors independently prove that the edge-weighted \(k\)-cardinality tree problem is \(\mathbb{NP}\)-hard: see [67,78,85,108,135]. In [108] it is shown that the problem is still \(\mathbb{NP}\)-hard if \(c(e) \in \{1, 2, 3\}\) for all edges \(e\) and \(G = K_n\), but polynomially solvable if there are only two distinct weights. The problem is polynomially solvable if \(G\) is a tree, see [107]. Another polynomial-time algorithm for the problem on trees can be found in [35].

The edge-weighted problem is \(\mathbb{NP}\)-hard for planar graphs and for Euclidean graphs, i.e. complete graphs, where the vertices are points in the plane and edge weights correspond to Euclidean distances, see [108].

In the same paper polynomial algorithms for decomposable graphs and graphs with bounded tree-width are given. (A graph is called decomposable if it is the join of at least two graphs, i.e. it consists of \(n \geq 2\) graphs \(G_1, G_2, \ldots, G_n\) and contains all the edges between the vertex sets of each pair of graphs \(G_i\) and \(G_j\) with \(i \neq j\). For the notion of bounded tree-width we refer the reader to the seminal paper of Robertson and Seymour [122].) A polynomial algorithm is also shown for the problem in the plane, when all points lie on the boundary of a convex region.

In [54], the authors focus on graded distance matrices. A symmetric distance matrix is called graded up (down) its rows if

\[ d_{ij} \leq (\geq) d_{il} \quad \text{for all } i < j \leq l \leq n. \]

The property of being graded up (down) the columns is defined in an analogous way. They assume that \(G = K_n\) and prove that \(k\)Card tree is \(\mathbb{NP}\)-hard on matrices graded up its rows or columns, whereas it is solvable in polynomial-time if the matrix is graded down its rows (columns) or both graded up its rows and columns.
For the vertex-weighted case, \( \mathbb{NP} \)-hardness is shown independently in [56,62]. Several authors consider special types of graphs. Note first that the vertex-weighted problem is trivially solved when \( G = K_n \). It is polynomially solvable if \( G \) is a tree, see [62]. It should also be noted that when \( G \) is a tree, the edge- and vertex-weighted versions are equivalent, see [59]. Polynomial time algorithms for the vertex-weighted problem exist for interval graphs and co-graphs [132].

Blum [22] and Blum and Ehrgott [25] describe another polynomially solvable special case. Given an undirected graph \( G = (V, E) \) and a vertex weight function \( w : V \rightarrow \mathbb{N} \), the authors define two kinds of connected subgraphs: troughs and hurdles. A trough is a connected subgraph \( S = (V(S), E(S)) \) of \( G \) such that

1. \( w(v_i) = w(v_j), \forall v_i, v_j \in V(S) \),
2. \( w(v) > w(v_i), \forall v_i \in V(S) \) and \( \forall v \in V \setminus V(S) \) adjacent to nodes in \( S \).

A hurdle is a connected subgraph \( H = (V(H), E(H)) \) of \( G \) such that

1. There exists a \( c \in \mathbb{R} \) with \( c \leq w(v_i) \forall v_i \in V(H) \).
2. There is at least one pair of nodes \( v_i, v_j \in V \setminus V(S) \) adjacent to nodes in \( H \) with \( w(v_i) < c \), \( w(v_j) < c \) and the path with smallest vertex weight from \( v_i \) to \( v_j \) in \( G \) contains at least one vertex \( v \in V(H) \).

The authors show that \( k \) Card tree is polynomially solvable, if \( G \) contains a single trough, or no hurdles.

On the other hand \( \mathbb{NP} \)-hardness results for the vertex-weighted case are obtained for grid and split graphs by Woeginger [132].

Concerning methodology, both exact and heuristic algorithms are developed, with a general focus on approximation algorithms. We first note that integer programming formulations are presented in [67,76]. Based on detailed studies of the associated polyhedron in the former paper a branch-and-cut algorithm is developed in [70]. The code and also implementations of most of the heuristics in [59] are documented [58], and are available as public domain software at http://optimierung.mathematik.uni-kl.de/project/index.html. A branch-and-bound method is described [40].

The heuristics mentioned are based on greedy and dual greedy (also called “stingy”) strategies and also make use of dynamic programming approaches. Some authors successfully apply local search methods to the \( k \)-cardinality tree problem: (Joernsten and Lokketangen [90]) apply a tabu search strategy, Catanas [37] presents both a genetic algorithm and a tabu search method. Yet another tabu search implementation is given by [20] with numerical results shown in Blesa and Xhafa [21]. A genetic algorithm is proposed in [50] and a memetic algorithm by Blesa et al. [19]. Other constructive heuristics are presented in [40]. Very recently, a new heuristic based on variable neighbourhood decomposition search (VNDS) is proposed in [127]. The method obtains solutions comparable to the state-of-art on test problem in literature (see http://iridia.ulb.ac.be/~cblum/kctlib/instances/index.html), whereas for larger problem instances up to 5000 vertices, VNDS is found to be superior in terms of quality of solutions and computation time. A heuristic-based on the ant colony optimization technique is proposed in [23]. The numerical results indicate that it performs very well.

A comparison of generic local search, genetic algorithms, and tabu search for the vertex-weighted problem is undertaken in [22], see also [25]. The results reported are often better than those obtained by constructive heuristics of Ehrgott et al. [59].

A large body of literature is available on approximation algorithms for the problem. The papers published on this topic represent an ongoing improvement until finally a constant approximation factor could be obtained. A first result appears in Woeginger [132]. He proves that there is an \( O(\sqrt{k}) \)-approximation algorithm for the vertex-weighted problem on grid graphs. Later there have been two streams of research articles, one focusing on the problem on general graphs \( G \), the other dealing with Euclidean graphs.

In the former, Marathe et al. [108] obtain a \( 2\sqrt{k} \)-approximation, which is improved in Awerbuch et al. [12] and Awerbuch et al. [13] to \( O(\log^2 k) \). A first constant factor approximation with factor 17 is derived in Blum et al. [26]. The constant factor approximation is 3, for the rooted case, in [76] and to 2.5, for the non-rooted case, in [9]. Arora and Karakostas [8], by modifying Garg’s 3-approximation algorithm, obtains a \( (2 + \varepsilon) \)-approximation algorithm with complexity \( n^{O(1/\varepsilon)} \) for any \( \varepsilon > 0 \), which represents the current best approximation guarantee for the \( k \)-cardinality tree which has been published.

For the problem in the plane, [108] gives an \( O(k^{1/4}) \)-approximation. Improved algorithms, with ratio of \( O(\log k) \) are due to Garg and Hochbaum [77], Rajagopalan and Vazirani [118], and Mata and Mitchell [110]. Further decrease to
O(log k/ log log n) [61] and a first constant factor approximation [24], with the constant not specified, follow. Using guillotine subdivisions, Mitchell [113], Mitchell [114] and Mitchell et al. [115] developed a 2√2-approximation for the l2 metric, and a 2-approximation for the l1 metric. Finally, a polynomial-time approximation scheme is given by [6]. The same author proposes nearly linear time approximation schemes that also work in higher dimensions, see [7].

In Kataoka et al. [94] and Kataoka and Araki [93] a rooted version of the edge-weighted problem is considered. Upper and lower bounding procedures and a Lagrangian relaxation approach are developed, respectively.

The following two sections are devoted to two problems which are closely related to the k-cardinality tree problem, namely the k-cardinality subgraph and travelling salesman problems.

3. The k-cardinality TSP and related routing problems

In this section we assume that the problems are defined on complete graphs. Because a k-cardinality tree can be used to construct a cycle containing exactly the k + 1 nodes of the subtree analogous to the problem without cardinality constraints (see e.g. [104]), many of the references cited in Section 2 have treated both k-cardinality tree as well as k-cardinality TSP.

Many authors observe that the problem is obviously \(\mathcal{NP}\)-hard, because, for \(k = n\) it is the TSP. We again distinguish between the planar and the general case.

For the problem where nodes are points in the plane, heuristics known from the TSP (r-opt and savings heuristic) are used for the k-cardinality TSP in a paper by Hamacher and Moll [86]. The paper also contains a geometric method based on clustering and a branch-and-bound algorithm. The details are in [83]. Much of the research is focused on approximation algorithms. Mata and Mitchell [110] present a constant factor approximation and [6] gives a polynomial-time approximation scheme.

Concerning the problem on general graphs, we mention [76], a 3-approximation. This is used to obtain a 10.77-approximation for the minimum latency problem (i.e. given \(n\) points \(v_1, \ldots, v_n\) find a tour starting at \(v_1\) and visiting all points, for which the sum of the arrival times is minimal). The ideas of the O(log^2 k)-approximation algorithm for kCard tree in [12] are used to obtain similar results for related problems: O(log^2 min(R, n)) for the quota-driven TSP and the prize-collecting TSP. In the quota-driven TSP an edge-weighted graph having integral values on the vertices is given and we want to find a route as short as possible that visits vertices whose total value is at least \(R\) (the salesman may visit a city several times but in this case he only receives its value once). The prize-collecting TSP differs from the previous problem since for each vertex a penalty value is specified besides the prize value: the goal is to find a tour of minimal “cost” such that the sum of prizes on the vertices visited is at least \(R\) where cost is defined to be the length of the tour plus the sum of the penalties on vertices not visited.

A similar bound for the orienteering problem is also given, i.e. the problem of maximizing the total value \(R\) of nodes visited, travelling at most a fixed distance \(d\).

4. The k-cardinality subgraph problem

This problem is perhaps even closer to the kCard tree problem than the k-cardinality TSP. The only change is that the requirement of acyclicity is dropped. Thus, the objective is to find a connected subgraph of \(G\) containing exactly either \(k\) edges or \(k\) vertices and having a minimal (maximal) total edge or vertex weight.

That all four variants of the (minimization) problem are \(\mathcal{NP}\)-hard is proved in a diploma thesis by Ehrgott [56]. An integer programming formulation, discussion of the polyhedral structure and a branch-and-cut algorithm for the problem can be found in [57]. The codes are available together with the above-mentioned codes for kCard tree [58].

Most authors have considered k-cardinality subgraph in the maximization version. When the cardinality constraint is put on the vertices, the subgraph is usually assumed to be induced by the set of selected vertices. A greedy algorithm with detailed analysis (including proofs of tight lower and upper bounds) of its worst-case performance ratio is proposed in [11]. Ravi et al. [119,120] consider the case \(G = K_n\) and two types of objectives, namely maximizing the minimal weight (bottleneck objective) and maximizing the sum of the weights between the selected vertices. For the bottleneck objective they show that if the weights do not satisfy the triangle inequality, there is no polynomial-time fixed-ratio approximation algorithm, unless \(\mathcal{P} = \mathcal{NP}\). If they do, a 2-approximation algorithm has been presented, as well as
a proof of \( \mathbb{NP} \)-hardness of obtaining a better performance ratio. For the sum problem, a 4-approximation is given, provided the triangle inequality is satisfied. A similar analysis is carried out in [100] for a minimization of the largest weight (bottleneck), the sum, and the variance of the weights. \( \mathbb{NP} \)-hardness in general, non-existence of fixed-ratio approximation algorithms in the absence of the triangle inequality and \( \mathbb{NP} \)-hardness of approximation with a ratio less than 2 in its presence are proved. Moreover, for trees, a polynomial-time algorithm is developed.

Kortsarz and Peleg [97] observed that the problem is \( \mathbb{NP} \)-hard, even if \( w(e) = 1 \) for all edges \( e \). They prove an \( O(n^{0.3885}) \)-approximation for the problem, which they call the densest \( k \)-vertex subgraph problem (i.e. find a \( k \)-vertex subgraph with the maximum number of edges).

\( \mathbb{NP} \)-hardness of this problem when the maximum degree is 3 is proved in [65]. The authors also give an algorithm which finds a subgraph with at least \( (1 - \varepsilon)^k \) edges in \( n^{O((1 + \log n/k)/\varepsilon)} \) time. In [63] an approximation algorithm with ratio \( O(n^{1/3}) \) for some \( \delta < 1/3 \) is given.

A problem with cardinality constraints on both vertex and edge set is considered by Asahiro and Iwama [10]. The authors give bounds on performance ratios for the feasibility problem.

A more general setting is discussed by Nehme and Yu [116], where a subset of \( k \) vertices of a hypergraph is to be found, maximizing the weight of the induced subhypergraph and subject to additional precedence constraints. For several special cases polynomial-time algorithms, \( \mathbb{NP} \)-hardness results are given.

In [79] the cardinality constraint is put on the edges and the weights on the vertices. \( \mathbb{NP} \)-hardness is proved even for \( w(v) = 1 \) for all vertices \( v \) or for vertex degrees not bigger than 3. For the problem of finding a maximal weight subgraph with \( k \) edges an \( O(kn) \)-time 3-approximation algorithm is given as well as an \( O(n + m) \)-time 2-approximation for the unweighted case. Again, for the case of \( G \) being a tree the problem can be solved in polynomial-time. Note that this fact also follows from results mentioned in Section 2.

5. Graph partitioning problems with a cardinality constraint

In this subsection we consider various graph-partitioning problems having a cardinality constraint. Let \( G = (V, E) \) be an undirected graph and let \( w : E \rightarrow \mathbb{N} \) be a positive integral weight function on the edge set.

The max \( k \)-cut (min \( k \)-cut) problem is to find a partition of the vertex set \( V \) into \( k \) disjoint subsets \( V = (V_1, V_2, ..., V_k) \) such that the sum of the weights of the crossing edges

\[
\sum_{1 \leq r < s \leq k} \sum_{i \in V_r, j \in V_s} w([v_i, v_j])
\]

is maximized (minimized). Both the max version and the min version of this problem are \( \mathbb{NP} \)-hard (the latter for \( k > 2 \)). For max \( k \)-cut, Frieze and Jerrum [71] develop an approximate algorithm with a factor \( 1/(1 - 1/k + 2k^{-2}\ln k) \) via a semidefinite programming relaxation that generalizes the approach proposed in [78] for the max cut problem. For the min \( k \)-cut problem Saran and Vazirani [123] present an approximation algorithm with a factor \( 2 - 2/k \).

The \( k \)-partitioning or graph equipartition problem differs from the min \( k \)-cut only by the additional requirement that the subsets of the partition must have almost the same size, i.e. \( ||V_r| - |V_s|| \leq 1 \), for all \( r \) and \( s \) in \( \{1, \ldots, k\} \). This problem is \( \mathbb{NP} \)-hard since for \( k = 2 \) we obtain the well-known equicut problem. The \( k \)-partitioning problem has several applications, especially in circuit layout, see [106]. This justifies the variety of articles describing heuristics for this problem, see e.g. [33,41,89,96]. Donath and Hoffman [52] introduce eigenvalue methods to obtain lower bounds for the \( k \)-partitioning problem, which are further improved by Rendl and Wolkowicz [121] using semidefinite programming. Karisch and Rendl [92] show that a different semidefinite model combined with a linear cutting plane approach leads to very tight approximation of equipartition that allows to handle graphs with up to 200 vertices on an ordinary personal computer. Finally, Beznuckov et al. [17,18] introduce a new lower bound method based on the solution of an extremal set problem which is called the edge-isoperimetric problem. The method is applied to the equipartition of Hamming graphs, i.e. the Cartesian product of complete graphs.

A generalization of the previous two kinds of graph partitioning problems is the max (min) \( k \)-cut with given size of the parts, i.e. a max (min) \( k \)-cut problem where also the cardinality of each subset \( V_r \) of the partition is constrained to be equal to a fixed value \( t_r \) for \( r = 1, \ldots, k \). Ageev and Sviridenko [1] present a 1/2-approximation for the maximization version of this problem. For the minimization version no constant-factor algorithm is known for general graphs; however, when the edge weights satisfy the triangle inequality and \( k \) is fixed, Guttmann-Beck and Hassin [81] present
a polynomial-time 3-approximation algorithm. Using semidefinite programming Feige and Langberg [64] develop an approximation algorithm with ratio of at least $1/2 + \varepsilon$ independently from the cardinality of the parts, when $k = 2$. A problem closely related to the min-$k$-cut with given size of the parts is the metric minimum partitioning that is to partition a given finite metric space in subsets having fixed cardinalities so as to minimize the sum of the distances across the cut: for this problem [66] design different PTASs based on a hybrid placement method.

Another interesting problem in graph partitioning is the $k$-cardinality cut problem ($k$Card cut). It consists in finding a cut, i.e. a partition of the vertex set $V = V_1 \cup V_2$ such that $C := \{v_i, v_j : v_i \in V_1, v_j \in V_2\}$ has cardinality $k$ and minimal weight $\sum_{e \in C} w(e)$. This problem is treated for the first time in [31]. For general graphs the existence version of this problem is already strongly $NP$-complete, since the largest $k$ value for which $k$Card cut is feasible represents the solution of the simple max cut problem, which is strongly $NP$-complete, see [74, p. 210]. Without the cardinality constraint the $k$Card cut problem reduces to the min cut problem which can be solved efficiently.

Despite the $NP$-completeness result, there are some special cases for which the $k$Card cut problem can be solved in polynomial-time. One is the unweighted problem, if $G$ is a complete graph. Then there exists a cut containing $k$ edges if and only if $k = \lceil (n - i) \rceil$ for some $i \in \{1, \ldots, \lfloor n/2 \rfloor \}$. Another polynomially solvable case is obtained if $G$ is a tree. Since every subset of the edge set is a cut, one can simply select the edges with the $k$ smallest weights and define a cut appropriately. A random pseudo-polynomial algorithm is obtained if $G$ is a planar graph by reducing the $k$Card cut problem to the exact perfect matching problem in a suitable transformation of the geometric dual of $G$, see [32].

In the same paper tight lower bounds are obtained via a semidefinite programming relaxation (SDP) strengthened through triangle inequalities. Finally, a randomized approximation algorithm with a factor $1.138$ using the random hyperplane technique of Geomans and Williamson [78] is developed based on the SDP relaxation. The computational results presented in [32] reveal that for every instance of the problem for which the optimal value is known this algorithm finds an optimal solution.

6. Other fixed cardinality problems on graphs

We could only find a few references to other graph theoretical optimization problems with cardinality constraints. We mention [112], which to the best of our knowledge is the historically first appearance of a fixed cardinality combinatorial optimization problem, namely the weighted matching problem. A necessary and sufficient condition for a $k$-cardinality matching to be optimal is given. A detailed investigation of the $k$-cardinality assignment problem is done in [46]. The authors develop an efficient primal algorithm using preprocessing techniques. Recently a more efficient algorithm for the case of sparse cost matrix is developed by Dell’Amico et al. [48]. Extensive computational tests show that the new code is considerably faster, and efficiently solves very large sparse and dense instances.

The problem of finding a minimal weight $k$-cardinality clique in a complete graph is considered in [125]. Simple exchange heuristics are presented and numerical results reported.

Another problem related to $k$-cardinality clique is the $k$-clique covering problem. It consists in finding the minimum number of cliques of size $k$ such that all edges are covered by the cliques. This problem is $NP$-complete since for $k = 3$ we obtain the triangle partition problem which is $NP$-complete [88]. It can be viewed as a special case of the set covering problem and specialized approximation algorithms are developed in [80].

The problem of finding a minimal weight $k$-cover of all vertices by edges of an undirected graph is considered, in White [129]. The author develops an efficient algorithm exploiting the fact that the family of minimum forest covers of varying cardinality is related to the minimum spanning tree of the graph.

In [39] the $k$-shredder problem is considered. It consists in finding a set of $k$ vertices whose removal disconnects the graphs into at least three connected components. The authors present an $O(k^2 n^2 + k^3 n^{1.5})$ algorithm for finding all the $k$-shredders of a $k$-connected graph.

An other interesting problem, defined for directed graphs, is the $k$-constrained shortest path problem: given two vertices $s$ and $t$, it consists in finding a shortest directed path from $s$ to $t$ with at most $k$ arcs. This problem is polynomially solvable using dynamic programming (see [74, p. 214]). Another practical approach, investigated by Ahuja and Magnanti [4], is the Lagrangian relaxation of cardinality constraint in order to obtain shortest path subproblems. Dahl and Realfsen [44] show that for a restricted class of acyclic digraphs having important application in curve approximation (2-graphs) the problem has a linear programming formulation allowing to solve within a reasonable time bigger instances than those which can be faced through the two previous approaches. Alon et al. [5] instead tackle the problem of finding...
both in undirected and directed graphs a path of cardinality exactly $k$. Contrary to the previous problem this is an $\text{NP}$-complete problem. Using the color-coding randomized technique [5] obtain for directed graphs an algorithm with expected worst-case time either $m2^{O(k)}$ or $m2^{O(k)} \log(n)$ or $O(mk!)$; whereas for undirected graphs $m$ in the above bounds can be replaced by $n$, improving the expected worst-case time of a previous algorithm developed by Bodlaender [27].

Finally, in [82] the following rather general problem is discussed. Find a subset of $k$ nodes such that the weight of the minimal spanning tree (minimal Hamiltonian cycle, Steiner tree) on this set is maximized. For complete graphs these problems are $\text{NP}$-hard. The spanning tree and TSP version cannot be approximated within a factor of $n^{1/6}$ unless $\text{P} = \text{NP}$. A $(k - 1)$-approximation algorithm is given for both. On metric graphs there are a 4-approximation for tree, a 3-approximation for TSP and Steiner tree, which are tight. Approximation is $\text{NP}$-hard for a factor less than 2 (tree and TSP) and $\frac{4}{3}$ (Steiner tree), respectively. In the plane, approximation factors 2.25 (tree), 2.16 (Steiner tree) are obtained.

### 7. Location problems

A classical field in which cardinality constraints are imposed is location theory: choose among a given set of points (vertices of a graph) a subset of a given cardinality, such that some objective is maximized or minimized. In fact, several of the papers listed in Sections 4 and 6 also belong to this category.

In [124] four heuristics are compared empirically for the problem of selecting $k$ out of $m > k$ points $x_1, \ldots, x_m \in \mathbb{R}^s$ such that their total distance to their median location is minimized:

$$\min_{C \subseteq P \mid |C| = k} \min_{y \in \mathbb{R}^s} \sum_{x_i \in C} w_i d(x_i, y),$$

where $P = \{x_1, \ldots, x_m\}$.

In [2] the selection of such points under some objectives related only to the distance of the selected points is investigated. In particular the authors present polynomial-time algorithms for the $k$-diameter, $k$-variance, the smallest square and the smallest rectangle problems. The $k$-diameter problem consists in minimizing the maximum distance between any two points of the $k$ selected points. The $k$-variance is the problem of minimizing the sum of squares of the distances of all pairs of the selected points. The smallest square and the smallest rectangle consist in minimizing the perimeter of the axes-parallel square and rectangle, respectively, which contain the selected points. The general idea behind their algorithms is to use higher-order Voronoi diagrams (i.e. a partition of the plane in polygons associated with each point $p \in P$ in such a way that each polygon is formed by the points that are closer to $p$ than any other point of $P$). A similar approach is used in Dobkin et al. [51] for finding a smallest convex polygon containing at least $k$ points but Aggarwal et al. [2] add several new features. The problem of selecting $k$ points for which the convex hull has the minimum area is also faced by Eppstein [60]: he improves the results of Dobkin et al. [51] and Aggarwal et al. [2] replacing Voronoi diagrams with sets of nearest neighbors to the input points. In this way he obtains a $O(n^2 \log n)$ algorithm which improves by almost a factor $n$ the best previous performance.

A problem opposite to the $k$-diameter problem, namely to select $k$ points that maximize the minimal distance between any two points, is studied in [130]. The authors develop a heuristic which achieves an approximation factor of 3 in general and of 2 for certain $k$. In [120] it is shown that no approximation ratio better than 2 can be obtained for this problem. In [38] the problem is extended considering different objective functions for measuring the remoteness of the points. For all the problems the authors present algorithms and hardness results. In particular they provide the first nontrivial performance guarantee for the problem of locating a set of $k$ points such that the sum of their distances to their nearest neighbor in the set is maximized. The performance ratio guarantee is 2.

In our context of combinatorial optimization we have to mention the $k$-median and $k$-center problem on networks, and the $k$-facility location problem as a discrete location problem. In the $k$-median problem we require that at most $k$ facilities be opened and the total service cost measured as the sum of the distances of each client to the nearest open facility, be minimum. The $k$-center problem consists in finding $k$ service centers that minimize the maximum distance of a customer to a closest center.

Because excellent surveys on both the network and discrete multifacility location problem exist, we refer the interested reader to these, see [101–103] and [43] and references therein.
Yamaguchi et al. [133] and Hamacher and Schöbel [87] consider the problem to locate a cycle with given cardinality $k$ in a network such that the distance between cycle and nodes is minimized. For general graphs the problem is $\mathbb{NP}$-hard, while it can be solved in polynomial-time if $G$ is a grid graph.

8. Packing problems with cardinality constraints

Babel et al. [15] consider the $k$-partitioning problem. A set of $n = km$ items, each having a non-negative weight, is to be partitioned into $m$ subsets of exactly $k$ items, such that the largest weight of all subsets is minimal. The problem is $\mathbb{NP}$-hard. Approximation algorithms, the best having a performance bound of $4/3$, are discussed.

Dell’Olmo et al. [49] consider various uniform $k$-partitioning problems which consist in partitioning $m$ sets each of cardinality $k$, into $k$ sets of cardinality $m$ such that each of these sets contain exactly one element from every original set. The problems differ according to the particular measure of “set uniformity” to be optimized: most of them are polynomial and corresponding algorithm are provided, whereas few of them are proved to be $\mathbb{NP}$-hard.

A problem closely related to $k$-partitioning is the cardinality constrained $P || C_{\text{max}}$ scheduling problem which is a generalization of the classical $P || C_{\text{max}}$ problem (i.e. assign $n$ jobs to $m$ identical parallel machines in such a way that the maximum completion time of a job is minimized) where a given limit $k$ is imposed on the number of jobs that can be assigned to any machine. Therefore $k$-partitioning can be seen as the special case of the cardinality constrained $P || C_{\text{max}}$ problem arising when $n = mk$ and vice versa an instance of the former problem can be transformed into an equivalent instance of the latter by adding $mk - n$ dummy jobs with zero processing time.

For this strongly $\mathbb{NP}$-hard problem, Dell’Amico and Martello [47] develop adaptation of upper and lower bounds for $P || C_{\text{max}}$ and analyze their worst-case behavior showing that generally the cardinality constraint makes worse the performance of these bounds. For this reason they also introduce new specifically tailored lower bounds. Heuristics algorithms and a scatter search method for the same problem appear in [45]. For the particular case of the $P || C_{\text{max}}$ problem with two-machines, Tsai [126] develops a heuristic which is asymptotically optimal when job processing times are independent and uniformly distributed, whereas Bramel et al. [30] show that a modified version of the well-known Longest Processing Time First (LPT) heuristic has a worst-case ratio of $7/6$ and this bound is tight. Again for the two-machines case but with the objective to minimize the total weighted completion time Yang et al. [134] develop a semidefinite programming-based approximation algorithm with a worst-case ratio at most $1.1626$.

A problem closely related to the cardinality constrained $P || C_{\text{max}}$ is the $k$-cardinality bin-packing problem where like in the previous problem we want to assign at most $k$ items (jobs) to each bin (machine) so as to minimize the number of bins used without exceeding the capacity of the bins. A section of [42] is dedicated to this $\mathbb{NP}$-hard problem. An $O(n \log^2 n)$-time $3/2$ worst-case ratio algorithm is presented in Kellerer and Pferschy [95] which improves the performance ratio of the first algorithms proposed by Krause et al. [98] and [99]. The online version of the same problem is investigated in Babel et al. [14]. This paper contains an approximation algorithm, the performance ratio of which tends to $2$, as $k$ increases.

Kellerer and Pferschy are also involved in a study of the $k$-cardinality knapsack problem: Caprara et al. [36] introduced a linear storage, polynomial-time approximation scheme and a dynamic programming based fully polynomial-time approximation scheme for this problem. Mastrolilli and Hutter [109] give an improved FPTAS for the cardinality constrained knapsack problem.

9. Conclusion

In this paper we have summarized the state of the art of research on combinatorial optimization problems with a cardinality constraint. These problems emerge in several applications and have therefore gained some attention in recent years. We have shown that there is a rich literature on some such problems, especially the $k$-cardinality tree problem. The study of other $k$CardCOP is in progress. We apologize for possible works on $k$CardCOP not included in this survey because of our limited knowledge and would be grateful for any further references.

We notice that few general properties of this family of problems are known, as for example a necessary and sufficient condition for the integral property or the interrelation between the complexity of the bottleneck problem and the existence problem. To discover other general properties of this class of problems is still an open challenge.
References


