Minimal instances of database schemes with additive cardinality constraints

Konrad Engel and Sven Hartmann

Fachbereich Mathematik, Universität Rostock, 18051 Rostock, Germany

Abstract. In the entity-relationship approach to database design, integrity constraints are used to express database semantics. Cardinality constraints are a frequently used way of imposing restrictions to the structure of the databases. In this paper, we consider additive cardinality constraints which specify lower and upper bounds on the total number of relationships an entity of a fixed type may be involved in. It is natural to ask for the smallest fully-populated database that yields a given set of integrity constraints. For additive cardinality constraints this proplem is proved to be NP-complete. However, the problem can be tackled via branch & bound. To solve the relaxed optimization problem we use min-cost flow methods in a suitable network depending on the given database scheme.

1 Introduction

The motivation of this study is the entity-relationship model (ERM) introduced by Chen [3]. Within the recent years, his approach has become one of the most popular methods in high-level database design. It provides a convenient, powerful and implementation independent description of objects to be stored in a database. The ERM and its variations are frequently used in the conceptual design of database applications, and lots of data design tools employ its concepts.

Let us briefly summarize the basic concepts of the ERM. For a complete definition we refer to [3, 10]. Let V be a non-empty, finite set. In the context of the entity-relationship approach, the elements of V are called *entity types*. With each entity type v we associate a set $\mathrm{dom}(v)$ called the domain of v. Members of the sets $\mathrm{dom}(v)$ are *entities* of type e. In the sequel, the domains shall always be pairwise disjoint. Intuitively, entities can be seen as individuals, which are of interest for some application or purpose. By classifying them and specifying their significant properties, we obtain entity types, which are used to model the individuals in their domains.

Whenever two entity types v and w are related to each other, this can be expressed by a *relationship type* e connecting v to w. Relationship types are used to model relations between individuals, i.e. entities. A *relationship* of type e is defined as the image of e under a function assigning items from $\mathrm{dom}(v)$ and $\mathrm{dom}(w)$ to v, resp. w. With every relationship type e we associate a multiset $\mathrm{pop}(e)$ of relationships and call it the population of e.

Let E be a set of relationship types defined on V. Further, put $\mathrm{dom}(V) = \cup_{v \in V} \mathrm{dom}(v)$ and $\mathrm{dom}(E) = \cup_{e \in E} \mathrm{dom}(e)$. The pair $\mathfrak{G} = (V, E)$ is called a *database scheme* and $\mathfrak{H} = (\mathrm{dom}(V), \mathrm{pop}(E))$ an *instance* of $\mathfrak{G}$. From the graph–theoretical point of view, both $\mathfrak{G}$ and $\mathfrak{H}$ can be considered as graphs, where the database scheme $\mathfrak{G}$ is the homomorphic image of each of its instances $\mathfrak{H}$ under a suitable mapping.

Usually, database schemes come along with certain restrictions that limit the possible combinations of entities participating in relationships. These restrictions are called integrity constraints. They are frequently used to express the different ways by that data are associated with one another in a database. In database design, great attention is devoted to this modeling of semantics. Since many different associations may exist, various kinds of integrity constraints are considered in the literature (cf. Thalheim [9]). Among the most popular ones, there is the class of cardinality constraints which impose restrictions on the number of relationships an entity of a given type may be involved in. Cardinality constraints have already been discussed for a longer period. A general approach towards cardinality constraints is developed in [10]. Combinatorial aspects of these contraints have been considered e.g. in [7, 8, 4, 6].

In this paper, we will be concerned with a particular class of cardinality constraints, namely additive cardinality constraints. An additive cardinality constraint $\mathrm{comp}_{\mathrm{add}}(v) = (a, b)$ specifies that in each admissible instance of a database scheme $\mathfrak{G}$ every entity of type v participates in at least a and at most b relationships. Unlike original cardinality constraints, additive ones bound the total number of relationships an entity may be involved in. Note, that a can be chosen from $\mathbb{N} = \{0, 1, 2, \ldots\}$ and b from $\mathbb{N}_+ = \{1, 2, \ldots\}$.

Given additive cardinality constraints for each entity type in the database scheme $\mathfrak{G} = (V, E)$ by two functions a and b defined on the V, we are looking for an instance of $\mathfrak{G}$ satisfying each of the constraints. Such an instance shall be called admissible (w.r.t a and b). Obviously, the empty database is always a legal instance. However, in database design we are usually interested in *fully-populated* instances where each of the domains $\mathrm{dom}(v)$ $(v \in V)$ is non-empty.

Example: Our example deals with partnerships and common research activities. Assume there shall be initiated a network consisting of a certain number of universities and private research institutes. They will organize a catalogue of multilateral activities as for instance workshops and projects. In addition, the universities offer

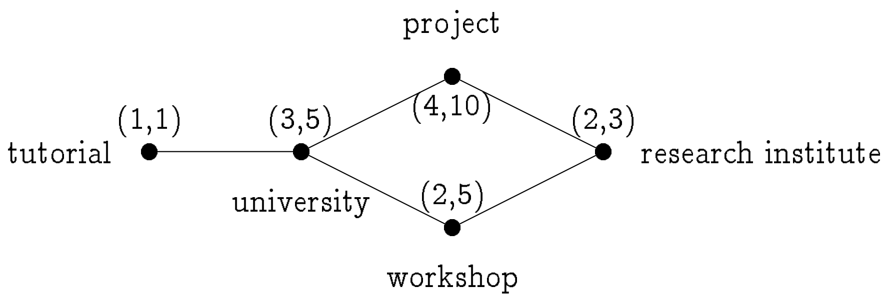

Figure 1: The database scheme with given additive cardinality constraints.

tutorials in the field of common interest. Using the entity-relationship approach, we obtain the database scheme shown in figure 1 with the vertices 'university', 'research institute', 'workshop', 'project' and 'tutorial'. However, due to financial and time limitations there is a number of restrictions to be considered. Each university shall be involved in at least 3 and at most 5 activities (workshops, projects, tutorials), each research institute in at least 2 and at most 3 activities (workshops or projects). On the other hand, each workshop shall be organized by at least 2 and at most 5 partners (universities or institutes), each tutorial by exactly one university. In each project, at least 4 and at most 10 partners shall be involved in. Note, that all these restrictions are additive cardinality constraints.

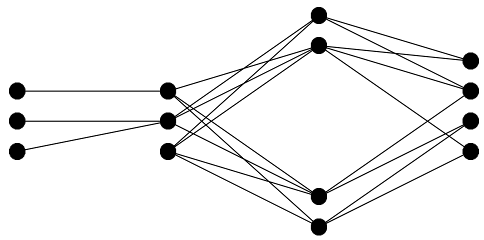

Figure 2: A fully-populated admissible instance of the database scheme.

A set of integrity constraints for a database scheme $\mathfrak{G}$ is said to be *consistent* iff they admit at least one fully-populated, admissible instance of $\mathfrak{G}$. Obviously, consistency is a basic requirement for the correctness of chosen constraints. As we will see later, additive cardinality constraints are always consistent. Thus, the following question naturally arises: Given a set of additive cardinality constraints, what is the smallest fully-populated legal instance of $\mathfrak{G}$. Knowing such a minimal instance provides us a lower bound on the size of the databases to be handled, and thus, enables us to decide whether the chosen representation of the modeled real world is managable or not.

This paper is organized as follows. In section 2, we reformulate the problem of finding instances in the context of graph theory. We will show that it is always possible to construct a fully-populated instance of a given database scheme, i.e., that each set of additive cardinality constraints is consistent. However, the problem of constructing a minimal fully-populated admissible instance turns out to be NP-complete. In section 3, we will provide a branch & bound algorithm where each relaxed problem can be solved using min-cost flow methods. Finally, in section 4 we will discuss our problem for a particular kind of additive cardinality constraints.

2 Graph–theoretical formulation of the problem

Let $\mathfrak{G} = (V, E)$ be a (simple) graph without isolated points and $a : V \to \mathbb{N},\ b : V \to \mathbb{N}_+$ be two functions such that $a(v) \leq b(v)$ for all $v \in V$. The problem consists in finding a graph $\mathfrak{H} = (W, F)$ (multiple edges are allowed) such that the following conditions are satisfied:

1. There exists a surjective mapping $f : W \to V$ such that $ww' \in F$ implies $f(w)f(w') \in E$ for all $w, w' \in W$ (i.e., $\mathfrak{G}$ is a homomorphic image of $\mathfrak{H}$, f is a *homomorphism*),
2. for all $w \in W$, the degree $d(w)$ of w is bounded as follows:

$$a(f(w)) \leq d(w) \leq b(f(w)),$$

3. the graph $\mathfrak{H}$ has minimum number of vertices, i.e., $|W| = \min$.

We call every graph $\mathfrak{H}$ satisfying the first two conditions an *instance of* $(\mathfrak{G}, a, b)$ and if, in addition, the third condition is satisfied, then $\mathfrak{H}$ is called a *minimal instance* of $(\mathfrak{G}, a, b)$. For a given instance $\mathfrak{H}$ with associated homomorphism f let

$$\begin{aligned} f^{-1}(v) &:= \{w \in W : f(w) = v\}, v \in V, \\ f^{-1}(vv') &:= \{ww' \in F : f(w) = v, f(w') = v'\}, vv' \in E, \\ x(v) &:= |f^{-1}(v)|, v \in V, \\ y(e) &:= |f^{-1}(e)|, e \in E. \end{aligned}$$

From the second condition we derive that for all $v \in V$,

$$\sum_{w \in f^{-1}(v)} a(v) \leq \sum_{w \in f^{-1}(v)} d(w) \leq \sum_{w \in f^{-1}(v)} b(v),$$

i.e.,

$$a(v)x(v) \leq \sum_{e \in E: v \in e} y(e) \leq b(v)x(v) \text{ for all } v \in V. \tag{1}$$

Of course,

$$x(v) \geq 1 \text{ for all } v \in V, \tag{2}$$

$$y(e) \geq 0 \text{ for all } e \in E, \tag{3}$$

and $x(v), y(e)$ are integral for all $v \in V, e \in E$. Note also that for the objective function $|W|$ we have

$$\sum_{v \in V} x(v) \to \min. \tag{4}$$

The converse is also true:

Theorem 1 *Let* $\mathfrak{G} = (V, E)$ *be a graph and let* $x, y : V \to \mathbb{N}$ *be two functions satisfying (1) – (3). Then there exists an instance* $\mathfrak{H} = (W, F)$ *of* $(\mathfrak{G}, a, b)$ *with some associated homomorphism* f *such that*

$$x(v) = |f^{-1}(v)| \text{ for all } v \in V, \tag{5}$$

$$y(e) = |f^{-1}(e)| \text{ for all } e \in E. \tag{6}$$

Proof: For each $v \in V$ let $W_v := \{w_{v,1}, \dots, w_{v,x(v)}\}$, where the sets W_v are pairwise disjoint. Moreover, let

$$W := \bigcup_{v \in V} W_v.$$

Now we will construct the edge set F of $\mathfrak{H}$ such that $f : W \to V$ defined by

$$f(w) := v \text{ iff } w \in W_v$$

is the desired homomorphism. The construction is described by the following algorithm in which edges are added step by step. We use the notation $\mod(m, n)$ for that representative of the residue class modulo n containing m which belongs to $\{1, \dots, n\}$, i.e., $r = \mod(m, n)$ iff n divides $m - r$ and $r \in \{1, \dots, n\}$.

```
{Initialization}
F := ∅;
for v ∈ V do h(v) := 0 od;
{Construction}
for vv' ∈ E do
    for j := 1 to y(vv') do
        h(v) := mod (h(v) + 1, x(v));
        h(v') := mod (h(v') + 1, x(v'));
        F := F ∪ {w_{v,h(v)} w_{v',h(v')}};
    od;
od.
```

By construction, $\mathfrak{G}$ is a homomorphic image of $\mathfrak{H}$ and (5) and (6) are fulfilled. So only the second condition for an instance must be verified. Since the edges are added "cyclically" w.r.t. the corresponding vertex sets, in each step of the algorithm the degrees of the vertices of each fixed class W_v differ by at most one, in particular also after the last step. Hence, for all $v \in V$,

$$\left\lfloor \frac{1}{x(v)} \sum_{e \in E : v \in e} y(e) \right\rfloor \le d(w) \le \left\lceil \frac{1}{x(v)} \sum_{e \in E : v \in e} y(e) \right\rceil \text{ for all } w \in W_v,$$

and (1) implies

$$a(v) \leq d(w) \leq b(v) \text{ for all } w \in W_v.$$

$\blacksquare$

By Theorem 1 we immediately derive the consistency of each set of additive cardinality constraints given by functions a and b.

Corollary 1 *For any graph $\mathfrak{G}$ and any functions $a : V \to \mathbb{N},\ b : V \to \mathbb{N}_+$ there exists an instance $\mathfrak{H}$ of $(\mathfrak{G}, a, b)$.*

Proof: By Theorem 1 we only have to verify that there is an integral solution of (1) – (3). If $a(v) = 0$ for all $v \in V$, then we may put $x(v) := 1$ for all $v \in V$ and $y(e) := 0$ for all $e \in E$. Otherwise, let

$$\alpha := \prod_{v \in V : a(v) \neq 0} a(v).$$

Further, let for $v \in V, e \in E$,

$$\begin{aligned} x(v) &:= \begin{cases} \left\lceil \frac{\alpha d(v)}{b(v)} \right\rceil & \text{if } a(v) = 0, \\ \frac{\alpha d(v)}{a(v)} & \text{if } a(v) > 0, \end{cases} \\ y(e) &:= \alpha. \end{aligned}$$

Then

$$a(v)x(v) \leq \alpha d(v) \leq b(v)x(v) \text{ for all } v \in V,$$

i.e., (1) holds. The relations (2) and (3) are trivially satisfied and x, y are obviously integral. $\blacksquare$

By Theorem 1 the problem of constructing an optimal instance of $(\mathfrak{G}, a, b)$ reduces to the solution of the discrete linear programming problem (1) – (4) which we call the *Additive Cardinality Constraints Problem (ACCP)*. Unfortunately, this problem cannot be solved in polynomial time, unless $\mathcal{P} = \mathcal{NP}$.

Theorem 2 *The Additive Cardinality Constraints Problem is NP–complete.*

Proof: We have to show that the corresponding decision problem is NP–complete, cf. [1, p. 127]:

ACCP–decision: Do there exist integral x, y such that (1) – (3) as well as

$$\sum_{v \in V} x(v) \leq L \tag{7}$$

hold, where L is some integer (from the input)?

Clearly, this problem belongs to $\mathcal{NP}$, since for guessed x and y, the inequalities (1) – (3) and (7) can be tested in polynomial time. To prove the NP–completeness we proceed by reduction from Knapsack to Restricted Knapsack to ACCP-decision. Let us recall the Knapsack Problem which is well–known to be NP–complete (cf. [1, p. 137]):

Knapsack: Given natural numbers $a_1, \ldots, a_n, b$, does there exist a 0-1 vector $\boldsymbol{x}$ such that $\sum_{j=1}^{n} a_j x_j = b$?

It is obvious that we may suppose that $a_j \leq b$ for all j without disturbing the NP-completeness. Here we introduce as an intermediate problem the following one:

Restricted Knapsack: Given natural numbers $c_1, \ldots, c_n, K$ such that $\max\{c_1, \ldots, c_n\} \leq 2\min\{c_1, \ldots, c_n\}$, does there exist a vector $\boldsymbol{z} \in \mathbb{N}_+^n$ such that $\sum_{j=1}^{n} c_j z_j = K$?

First of all note that in the proof of the NP–completeness of Restricted Knapsack we may replace the condition $\boldsymbol{z} \in \mathbb{N}_+^n$ by the condition $\boldsymbol{z} \in \mathbb{N}^n$ since, for $\boldsymbol{z} \in \mathbb{N}_+^n$, $\sum_{j=1}^{n} c_j z_j = K$ iff $\sum_{j=1}^{n} c_j (z_j - 1) = K - \sum_{j=1}^{n} c_j$. Now we will show that Knapsack $\propto$ Restricted Knapsack. Given an instance $(a_1, \ldots, a_n, b)$ of Knapsack (with $a_j \leq b$ for all j) we define an instance $(c_1, \ldots, c_{2n}, K)$ of Restricted Knapsack by

$$\begin{aligned}
M' &:= 2n(n+1)b, \\
M &:= 2^t \text{ where } t \text{ is the smallest integer with } 2^t > M', \\
K &:= nM^{n+1} + \sum_{j=1}^{n} M^j + b = nM^{n+1} + \frac{M^{n+1} - M}{M - 1} + b, \\
c_j &:= \begin{cases} M^{n+1} + M^j + a_j & \text{if } 1 \leq j \leq n, \\ M^{n+1} + M^{j-n} & \text{if } n < j \leq 2n. \end{cases}
\end{aligned}$$

This construction is similar to the construction presented e.g. in [2, 15.7]. Here really all numbers can be computed in polynomial time. In order to see this easier we replaced M' by M since then M^{n+1} can be calculated in a trivial way if the computations are carried out in the binary code. Note that

$$\frac{\max\{c_1, \ldots, c_{2n}\}}{\min\{c_1, \ldots, c_{2n}\}} < \frac{M^{n+1} + M^n + b}{M^{n+1}} = 1 + \frac{1}{M} + \frac{b}{M^{n+1}} \leq 2.$$

Now, if $(a_1, \dots, a_n, b)$ is feasible with corresponding vector x, then we may define

$$z_j = \begin{cases} x_j & \text{if } 1 \le j \le n, \\ 1 - x_{j-n} & \text{if } n < j \le 2n. \end{cases}$$

It is then easy to see that the constructed tuple $(c_1, \dots, c_{2n}, K)$ is feasible with corresponding vector z. Conversely, let the constructed tuple $(c_1, \dots, c_{2n}, K)$ from Restricted Knapsack be feasible with corresponding vector z. Then

$$\left(\sum_{j=1}^{2n} z_j\right) M^{n+1} + \sum_{j=1}^{n} (z_j + z_{n+j}) M^j + \sum_{j=1}^{n} a_j z_j = nM^{n+1} + \sum_{j=1}^{n} M^j + b. \tag{8}$$

Since $\sum_{j=1}^{2n} c_j z_j = K$, obviously, for all j,

$$z_j \le \frac{K}{c_j} < \frac{1}{M^{n+1}} \left(nM^{n+1} + \sum_{j=1}^{n} M^j + b \right) < n + 1 < \frac{M}{2nb}.$$

Hence,

$$\begin{aligned} \sum_{j=1}^{2n} z_j &< 2n \frac{M}{2nb} < M, \\ z_j + z_{n+j} &< \quad 2 \frac{M}{2nb} < M \text{ for all } j, \\ \sum_{j=1}^{n} a_j z_j &< \quad \frac{nbM}{2nb} < M, \end{aligned}$$

i.e., all coefficients of the LHS of (8) are smaller than M. Comparison of coefficients yields

$$\begin{aligned} \sum_{j=1}^{2n} z_j &= n, \\ z_j + z_{n+j} &= 1 \text{ for all } j, \\ \sum_{j=1}^{n} a_j z_j &= b. \end{aligned}$$

Thus $z_j \in \{0, 1\}$ for all j. The vector z shows that the instance $(a_1, \dots, a_n, b)$ is feasible.

It remains to prove that Restricted Knapsack $\propto$ ACCP–decision. Given an instance $(c_1, \ldots, c_n, K)$ of Restricted Knapsack we define an instance $(\mathfrak{G}, a, b, L)$ of ACCP-decision as follows: Let

$$\begin{aligned} V &:= \{1, \ldots, n, n+1\}, \\ E &:= \{\{1, n+1\}, \ldots, \{n, n+1\}\}, \\ a(j) &:= b(j) := c_j, \ j = 1, \ldots, n, \\ a(n+1) &:= b(n+1) := K, \\ L &:= \left\lceil \frac{K}{\min\{c_1, \ldots, c_n\}} \right\rceil + 1. \end{aligned}$$

Here the graph $\mathfrak{G} = (V, E)$ is a star with central point $n+1$. Clearly, this instance can be constructed in polynomial time. Let $(c_1, \ldots, c_n, K)$ be feasible with corresponding vector $z \in \mathbb{N}_+^n$. Then we define

$$\begin{aligned} x(j) &:= z_j, \ j = 1, \ldots, n, \\ x(n+1) &:= 1, \\ y(\{j, n+1\}) &:= c_j z_j. \end{aligned}$$

It is easy to see that (1) – (3) are satisfied and, with $\mu := \min\{c_1, \ldots, c_n\}$,

$$\sum_{j=1}^{n+1} x(j) = \sum_{j=1}^{n} z_j + 1 \leq \sum_{j=1}^{n} \frac{c_j z_j}{\mu} + 1 = \frac{K}{\mu} + 1 \leq L,$$

i.e., $(\mathfrak{G}, a, b, L)$ is feasible. Conversely, let the constructed $(\mathfrak{G}, a, b, L)$ be feasible with corresponding functions $x, y : V \to \mathbb{N}$. Assume that $x(n+1) > 1$. Since

$$\sum_{j=1}^{n} c_j x(j) = \sum_{j=1}^{n} y(\{j, n+1\}) = Kx(n+1),$$

we have, with the abbreviation $\nu := \max\{c_1, \ldots, c_n\}$,

$$\sum_{j=1}^{n} x(j) \geq \sum_{j=1}^{n} \frac{c_j x(j)}{\nu} = \frac{Kx(n+1)}{\nu} \geq \frac{2K}{\nu} \geq \frac{K}{\mu}.$$

It follows that

$$\sum_{j=1}^{n+1} x(j) \geq \left\lceil \frac{K}{\mu} \right\rceil + 2 = L + 1,$$

a contradiction. Consequently, we have $x(n+1) = 1$, and thus

$$\sum_{j=1}^{n} c_j y(\{j, n+1\}) = K.$$

Hence, the vector z with $z_j := y(\{j, n+1\})$ shows that $(c_1, \ldots, c_n, K)$ is feasible. ■

3 Exact solution via Branch & Bound

In general, the ACCP can be solved via any solver of integer linear programming problems (ILP). But because of the nice structure of the system of inequalities (1) – (3), for a large subclass, one can improve the efficiency essentially. The following observation shows that, under some weak assumptions, it is enough to force the variables $x(v), v \in V$, to be integral.

Theorem 3 *Let x, y be a solution of (1) – (3) and let x be integral. If $\mathfrak{G}$ is bipartite or if $a(v) < b(v)$ for all $v \in V$, then there exists some integral y' such that x, y' is a solution of (1) – (3).*

Proof: First let $\mathfrak{G}$ be bipartite. For fixed x, the system (1), (3) has the form

$$\begin{pmatrix} A \\ -A \end{pmatrix} \mathbf{y} \leq \mathbf{c}, \tag{9}$$

$$\mathbf{y} \geq 0, \tag{10}$$

where $\mathbf{c}$ is integral and A is the incidence matrix of $\mathfrak{G}$. By the supposition, the polytope described by (9), (10) is not empty. For $\mathbf{y}'$ (the vector with components $y'(e), e \in E$) we may take any vertex of this polytope since A is well–known to be totally unimodular and hence the polytope is integral.

Now let $\mathfrak{G}$ be arbitrary but $a(v) < b(v)$ for all $v \in V$. In a standard way we construct a bipartite graph $G_1 = (V_1 \cup V_2, E_1)$ whose color classes V_1, V_2 are copies of V and where $v_1 w_2 \in E$ iff $vw = wv \in E$ (here v_1, w_2 are the corresponding copies of v and w). If we put $y_1(v_1 w_2) := y(vw)$ for all $vw \in E$, then obviously

$$\begin{aligned} a(v)x(v) \leq \sum_{e \in E_1 : v_1 \in e} y_1(e) &\leq b(v)x(v) \text{ for all } v_1 \in V_1, \\ a(w)x(w) \leq \sum_{e \in E_1 : w_2 \in e} y_1(e) &\leq b(v)x(v) \text{ for all } w_2 \in V_2, \\ y_1(e) &\geq 0 \text{ for all } e \in E_1. \end{aligned}$$

By the preceding case, there exists some integral y_1' satisfying these inequalities. If we put

$$y''(vw) := \frac{1}{2}(y_1'(v_1 w_2) + y_1'(w_1 v_2)) \text{ for all } v, w \in E,$$

then y'' satisfies also (1) and (3). Let $G_2 = (V, E_2)$ be the subgraph of $\mathfrak{G}$ where

$$E_2 := \{e \in E : y''(e) \text{ is not integral (but then half–integral)}\}.$$

In a standard way we partition G_2 into vertex–disjoint closed walks and into some paths (which are not necessarily vertex–disjoint and may have also common inner points with the closed walks). Now we make y'' integral, i.e. construct y', by adding and substracting $\frac{1}{2}$ on the edges of E_2. On the closed walks of even length and on the paths we do this in any alternating way. Then (1) and (3) will not be violated (note that the lower and upper bounds in (1) are integral). On the closed walks $(v_1, \dots, v_k, v_1)$ of odd length we proceed also in an alternating way, but we have to say for a fixed edge v_1v_2 whether to start with adding or substracting $\frac{1}{2}$. We add $\frac{1}{2}$ iff $\sum_{v_1 \in e} y''(e) < x(v_1)b(v_1)$. By the integrality, also in this case (1) and (3) will not be violated (note that $\sum_{v_1 \in e} y''(e) = x(v_1)b(v_1)$ implies $\sum_{v_1 \in e} y''(e) > x(v_1)a(v_1)$ because of $a(v_1) < b(v_1)$). ■

The algorithmic aspects of constructing y' (resp. y_1' and hence y'') will be given in the sequel. Following the standard Branch & Bound algorithm for ILP (cf. [1, p. 356]) we solve the LP–relaxation of corresponding subproblems and branch always at some variable $x(v)$ where $\xi := x(v)$ is not integral. This leads to two subproblems where in addition

$$x(v) \leq \lfloor \xi \rfloor \text{ and } x(v) \geq \lfloor \xi \rfloor + 1, \tag{11}$$

respectively. Thus all subproblems have the following form:

Let $a : V \to \mathbb{N}$ and $b, \alpha, \beta : V \to \mathbb{N}_+$ be functions with $a(v) \leq b(v)$ and $\alpha(v) \leq \beta(v)$ for all $v \in V$. Find $x : V \to \mathbb{R}$, $y : E \to \mathbb{R}$ satisfying (1), (3), (4), and

$$\alpha(v) \leq x(v) \leq \beta(v) \text{ for all } v \in V. \tag{12}$$

We call this problem the *Relaxed Additve Cardinality Constraints Problem (RACCP)*. The main observation is the fact that RACCP can be solved efficiently with a min–cost–flow algorithm. Thus we have a new interesting example where at each node of the Branch & Bound Tree there is a *nicely solvable subproblem* and where the variables are *not only binary*.

Let $\mathfrak{G} = (V, E)$ be bipartite with color classes V_1, V_2. With $\mathfrak{G}$ we associate the (directed) network $N = (V_N, E_N, q, s, l, u, c)$ where $V_N := V \cup \{q, s\}$ (q, s are new vertices – the *source* and the *sink*), $E_N := E_1' \cup E_1'' \cup E_{12} \cup E_2' \cup E_2''$ with $E_1' := E_1'' := \{qv_1 : v_1 \in V_1\}$, $E_2' := E_2'' := \{v_2s : v_2 \in V_2\}$, $E_{12} := \{v_1v_2 : v_1 \in V_1, v_2 \in V_2\}$ (the edges of E become (directed) arcs from V_1 to V_2). Here the union is considered in the multiset–sense, i.e., we have parallel arcs between the source and the vertices of V_1 and between the vertices of V_2 and the sink. To indicate this we often denote arcs from $E' := E_1' \cup E_2'$ by e' and arcs from $E'' := E_1'' \cup E_2''$ by e''. Furthermore, l and u are *lower*, resp. *upper* *capacity functions* defined on E_N by

$$l(e) := \begin{cases} a(v)\alpha(v) & \text{if } (e = qv \text{ or } e = vs) \text{ and } e \in E', \\ 0 & \text{if } e \in E'' \cup E_{12}, \end{cases}$$

$$u(e) := \begin{cases} b(v)\alpha(v) & \text{if } (e = qv \text{ or } e = vs) \text{ and } e \in E', \\ b(v)(\beta(v) - \alpha(v)) & \text{if } e \in E'', \\ \infty & \text{if } e \in E_{12}. \end{cases}$$

Finally, c is a *cost function* defined on E_N by

$$c(e) := \begin{cases} 0 & \text{if } e \in E', \\ \frac{1}{b(v)} & \text{if } (e = qv \text{ or } e = vs) \text{ and } e \in E'', \\ 0 & \text{if } e \in E_{12}. \end{cases}$$

For $e = vw$ let $e^- := v$ and $e^+ := w$. A *flow* on N is a function $f : E_N \to \mathbb{R}$ such that

$$\sum_{e^+=v} f(e) = \sum_{e^-=v} f(e) \text{ for all } v \in V.$$

We call the flow *admissible* if in addition

$$l(e) \le f(e) \le u(e) \text{ for all } e \in E_N.$$

The *cost of the flow* f is defined by

$$c(f) := \sum_{e \in E_N} c(e)f(e)$$

which is, for the network constructed above, equal to $\sum_{e \in E''} c(e)f(e)$. A solution of the Relaxed Additve Cardinality Constraints Problem can be obtained via the following theorem and any min–cost–flow algorithm.

Theorem 4 *Let f^* be an admissible flow on N of minimum cost. Then an optimal solution of RACCP is given by*

$$x^*(v) := \alpha(v) + \frac{1}{b(v)} f^*(e''_v),$$

where e''_v is the unique arc of E'' being incident with v,

$$y^*(e) := f^*(e),$$

where $e \in E_{12}$ and on the LHS e is considered as an undirected edge.

If there is no admissible flow on N, then there is no admissible solution of RACCP.

Proof: It is an easy exercise to verify that x^* and y^* are admissible solutions of RACCP. As an example we prove only the first inequality in (1) for a vertex $v \in V_1$. We have

$$\begin{aligned}\sum_{e\in E: v\in e} y^*(e) = \sum_{e^-=v} f^*(e) = \sum_{e^+=v} f^*(e) &= f((qv)') + f((qv)'') \\ &\geq a(v)\alpha(v) + b(v)(x^*(v) - \alpha(v)) \\ &\geq a(v)\alpha(v) + a(v)(x^*(v) - \alpha(v)) = a(v)x^*(v).\end{aligned}$$

Note that the value of the objective function of RACCP (see (4)) equals

$$\sum_{v\in V} x^*(v) = \sum_{v\in V} \alpha(v) + \sum_{v\in V} \frac{1}{b(v)} f^*(e_v'') = \sum_{v\in V} \alpha(v) + c(f^*). \tag{13}$$

To show that we have an optimal solution of RACCP we will construct for every admissible solution x, y of RACCP an admissible flow f such that

$$c(f) \leq \sum_{v\in V} (x(v) - \alpha(v)). \tag{14}$$

Since f^* is a flow of minimum cost we have then by (13) and (14)

$$c(f^*) = \sum_{v\in V} x^*(v) - \sum_{v\in V} \alpha(v) \leq c(f) \leq \sum_{v\in V} x(v) - \sum_{v\in V} \alpha(v),$$

i.e.,

$$\sum_{v\in V} x^*(v) \leq \sum_{v\in V} x(v).$$

This construction then also shows that there is no admissible solution of RACCP if there is no admissible flow on N. Here is the construction: Let

$$h(v) := \sum_{e\in E: v\in e} y(e) - a(v)(x(v) - \alpha(v)) - b(v)\alpha(v).$$

We put

$$f(e) := \begin{cases} b(v)\alpha(v) + \min\{0, h(v)\} & \text{if } (e = qv \text{ or } e = vs) \text{ and } e \in E', \\ a(v)(x(v) - \alpha(v)) & \\ \qquad\qquad + \max\{0, h(v)\} & \text{if } (e = qv \text{ or } e = vs) \text{ and } e \in E'', \\ y(e) & \text{if } e \in E_{12}. \end{cases}$$

For $v \in V_1$ we have then

$$\begin{aligned}\sum_{e^+=v} f(e) - \sum_{e^-=v} f(e) &= b(v)\alpha(v) + \min\{0, h(v)\} \\ &\quad + a(v)(x(v) - \alpha(v)) + \max\{0, h(v)\} - \sum_{e\in E: v\in e} y(e) \\ &= b(v)\alpha(v) + a(v)(x(v) - \alpha(v)) + h(v) - \sum_{e\in E: v\in e} y(e) \\ &= 0,\end{aligned}$$

and analogously this holds for $v \in V_2$, hence f is indeed a flow. The inequality $l(e) \le f(e) \le u(e)$ is trivially satisfied for $e \in E_{12}$. For $e \in E'$ we have

$$\begin{aligned}l(e) &= a(v)\alpha(v) = \min\{b(v)\alpha(v), a(v)\alpha(v)\} \\ &= \min\{b(v)\alpha(v), a(v)(x(v) - x(v) + \alpha(v)\} \\ &\le \min\{b(v)\alpha(v), \sum_{e\in E: v\in e} y(e) - a(v)(x(v) - \alpha(v))\} = f(e) \\ &\le b(v)\alpha(v) = u(e),\end{aligned}$$

and for $e \in E''$ we have

$$\begin{aligned}l(e) &= 0 \le a(v)(x(v) - \alpha(v)) \\ &\le \max\{a(v)(x(v) - \alpha(v)), \sum_{e\in E: v\in e} y(e) - b(v)\alpha(v)\} = f(e) \\ &\le \max\{b(v)(\beta(v) - \alpha(v)), b(v)(x(v) - \alpha(v))\} \\ &= b(v)(\beta(v) - \alpha(v)) = u(e),\end{aligned}$$

hence f is indeed an admissible flow. Finally, the cost of f is given by

$$\begin{aligned}c(f) &= \sum_{e\in E''} c(e)f(e) = \sum_{v\in V} \frac{1}{b(v)}(a(v)(x(v) - \alpha(v)) + \max\{0, h(v)\}) \\ &\le \sum_{v\in V} \frac{1}{b(v)} \max\{b(v)(x(v) - \alpha(v)), \sum_{e\in E: v\in e} y(e) - b(v)\alpha(v)\} \\ &\le \sum_{v\in V} \max\{(x(v) - \alpha(v)), (x(v) - \alpha(v))\} = \sum_{v\in V} (x(v) - \alpha(v)),\end{aligned}$$

i.e., (14) holds, and the proof is complete. ■

Since all capacities are integral, standard min–cost–flow algorithms yield integral optimal flows, hence the function y^* constructed in Theorem 4 is always integral. Only the function x^* need not be integral, and this gives the necessity of branching into subproblems described in (11). Since the investigations obviously may be restricted to a polytope (bounded polyhedron), the finiteness of the Branch & Bound Algorithm is clear, and it terminates at some integral x^*.

Now we look at the case where $\mathfrak{G}$ is not necessarily bipartite. As in the proof of Theorem 3 we associate with $\mathfrak{G}$ the bipartite graph $G_1 = (V_1 \cup V_2, E_1)$ whose color classes V_1, V_2 are copies of V and where $v_1 w_2 \in E_1$ iff $vw = wv \in E$ (v_1, w_2 are the cprresponding copies of v and w). Moreover, we define the function a_1 on $V_1 \cup V_2$ by

$$a_1(v_1) := a_1(v_2) := a(v)$$

where v_1, v_2 are the two copies of v. The functions b_1, α_1, β_1 are defined analogously. The following proposition is obvious.

Proposition 1 *a) If x_1, y_1 are admissible solutions of RACCP on G_1, then the functions x, y defined by*

$$x(v) := \frac{1}{2}(x_1(v_1) + x_1(v_2)) \text{ for all } v \in V,$$
$$y(vw) := \frac{1}{2}(y(v_1 w_2) + y(w_1 v_2)) \text{ for all } vw \in E,$$

where v_1, v_2 and w_1, w_2 are the copies of v and w, respectively, are admissible solutions of RACCP on $\mathfrak{G}$.

b) If x, y are admissible solutions of RACCP on $\mathfrak{G}$, then the functions x_1, y_1 defined by

$$x_1(v_1) := \quad x_1(v_2) := \quad x(v) \text{ for all } v_1 \in V_1, v_2 \in V_2, \tag{15}$$
$$y(v_1 w_2) := y(w_1 v_2) := y(vw) \text{ for all } v_1 w_2, w_1 v_2 \in E_1, \tag{16}$$

where again v_1, v_2 and w_1, w_2 are the copies of v and w, respectively, are admissible solutions of RACCP on G_1.

Note that in this correspondence,

$$2 \sum_{v \in V} x(v) = \sum_{w \in V_1 \cup V_2} x_1(w),$$

and thus optimal solutions of RACCP on $\mathfrak{G}$ and on G_1 can be bijectively mapped (with the formulas of Proposition 1) onto each other. Consequently, we may use the preceding method for bipartite graphs (verified by Theorem 4) to solve RACCP on $\mathfrak{G}$. The Branch & Bound Algorithm leads to an optimal solution under the additional condition of integrality of $x(v), v \in V$. By the integrality of all flows the variables $y(e)$ will be integral or half–integral in this solution. At least in the case $a(v) < b(v)$ for all $v \in V$, an optimal solution of ACCP (having also integral $y(e)$ for all $e \in E$) can be constructed by the method in the proof of Theorem 3.

4 The case $a(v), b(v) \in \{0, 1\}$ for all $v \in V$

Since we supposed that $b(v) > 0$ for all $v \in V$ we have in this section throughout $b(v) = 1$ for all $v \in V$. If $\mathfrak{G}$ is bipartite we may use the preceding approach, but do not need branch. The optimal solution of RACCP in the first step (i.e. $\alpha(v) = a(v), \beta(v) = b(v) = 1$ for all $v \in V$) is obviously integral (see Theorem 4), i.e., we have an optimal solution of ACCP. Thus let $\mathfrak{G}$ be arbitrary. Let

$$W := \{v \in V : a(v) = 1\}.$$

We define a weight function f on E by

$$f(e) := \begin{cases} 2 & \text{if } e \subseteq W, \\ 1 & \text{if } |e \cap W| = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Let M be a maximum weighted matching in $\mathfrak{G}$ (which can be obtained by standard matching algorithms, cf. [1, p. 628 ff]). Let $U \subseteq W$ be the subset of W of those points which are not covered by M. For each $v \in U$ we choose some edge e_v being incident with v (recall that we suppose throughout that $\mathfrak{G}$ does not contain isolated points). Let $E' := M \cup \{e_v : v \in U\}$ and let $\mathfrak{G}' := (V, E')$. We denote the degree of a vertex v of $\mathfrak{G}'$ by $d_{\mathfrak{G}'}(v)$.

Theorem 5 *An optimal solution of ACCP for* $(\mathfrak{G}, a, b)$ *is given by*

$$x^*(v) := \max\{1, d_{\mathfrak{G}'}(v)\},$$
$$y^*(e) := \begin{cases} 1 & \textit{if } e \in E', \\ 0 & \textit{otherwise.} \end{cases}$$

Proof: It is trivial to check that the presented solution is indeed admissible. Since the matching M is maximum, each $e_v, v \in U$, is incident with one point covered by M. Hence the value of the objective function is given by

$$\sum_{v \in V} x^*(v) = |V| + |U|.$$

Now assume that x^*, y^* is not an optimal solution, and let x, y be an optimal solution. In particular,

$$\sum_{v \in V} x(v) < |V| + |U|. \tag{17}$$

Obviously, we have for each $e = vw$, $y(e) \in \{0, 1\}$ because for $y(e) \geq 2$ we may reduce $x(v)$ and $x(w)$ by one and still have an admissible solution which contradicts the optimality of x, y.

Let $E'' := \{e \in E : y(e) = 1\}$. Let v be any point which has in $\mathfrak{G}'' := (V, E'')$ degree greater than 1. Then every point w which is adjacent to v in $\mathfrak{G}''$ has degree 1 in $\mathfrak{G}''$ and belongs to W because otherwise we could set $y(e) := 0$ and reduce $x(v)$ (and $x(w)$ if $x(w) \geq 2$) by one, again a contradiction to the optimality of x, y. It is easy to see that

$$x(v) = \max\{1, d_{\mathfrak{G}''}(v)\} \text{ for all } v \in V.$$

From $\mathfrak{G}''$ we delete step by step edges which are incident with a vertex of degree greater than one (in the remaining graph). This procedure finally yields a matching M'' and a set U'' of isolated points (note $U'' \subseteq W$). Moreover, obviously,

$$|U''| = \sum_{v \in V} x(v) - |V|. \tag{18}$$

For the weight $f(M'')$ of the matching M'' we have

$$\begin{aligned} f(M'') &= 2|\{e \in M'' : e \subseteq W\}| + |\{e \in M'' : |e \cap W| = 1\}| \\ &= |W| - |U''|. \end{aligned} \tag{19}$$

Analogously, the weight $f(M)$ of the maximum matching M equals

$$f(M) = |W| - |U|. \tag{20}$$

From (18) – (20) we derive

$$\begin{aligned} \sum_{v \in V} x(v) &= |U''| + |V| = |V| + |W| - f(M'') \\ &\geq |V| + |W| - f(M) = |V| + |W| - |W| + |U|, \end{aligned}$$

i.e.,

$$\sum_{v \in V} x(v) \geq |V| + |U|,$$

a contradiction to (17). ■

References

[1] G.L. Nemhauser and L.A. Wolsey: *Integer and combinatorial optimization*, Wiley (New York, 1988).

[2] C.H. Papadimitriou and K. Steiglitz: *Combinatorial optimization: Algorithms and complexity*, Prentice–Hall (Englewood Cliffs, N.J., 1982).

[3] P. Chen: The Entity-Relationship Model: Towards a unified view of data, *ACM TODS* 1,1 (1984) 9-36.

[4] K. Engel und S. Hartmann: *Constructing instances of semantic entity relationship schemes*, Preprint 94/3, Universitt Rostock (1994).

[5] S. Hartmann: Graph-theoretic methods to construct entity-relationship databases, in: M. Nagl, *Graphtheoretic concepts in computer science*, LNCS 1017, Springer (Berlin, 1995) 131-145.

[6] S. Hartmann: *"Uber die Charakterisierung und Konstruktion von Entity-Relationship-Datenbanken mit Kardinalit"atsbedingungen*, Ph.D. thesis, Universit"at Rostock (Rostock, 1996).

[7] M. Lenzerini and P. Nobili: On the satisfiability of dependency constraints in Entity-Relationship schemata, *Information Systems* 15 (1990) 453-461.

[8] B. Thalheim: Fundamentals of cardinality constraints, in: G. Pernul and A.M. Tjoa (eds.), *Entity-relationship approach*, LNCS 645, Springer (Berlin, 1992) 7-23.

[9] B. Thalheim: *A survey on Database Constraints*, Reihe Informatik I-8, Universit"at Cottbus (Cottbus, 1994).

[10] B. Thalheim: *Fundamentals of Entity-Relationship Models*, Springer (Berlin, 1997).