Discrete Applied Mathematics 201 (2016) 86–104

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Facets of the axial three-index assignment polytope

Trivikram Dokka [a,*], Frits C.R. Spieksma [b]

[a] Department of Management Science, Lancaster University Management School, Lancaster, LA1 14X, United Kingdom

[b] ORSTAT, KU Leuven, Naamsestraat 69, B-3000, Leuven, Belgium

ARTICLE INFO

Article history:
Received 9 April 2014
Received in revised form 17 July 2015
Accepted 18 July 2015
Available online 10 August 2015

Keywords:
Three-dimensional assignment
Polyhedral methods
Facets
Separation algorithm

ABSTRACT

We revisit the facial structure of the axial 3-index assignment polytope. After reviewing known classes of facet-defining inequalities, we present a new class of valid inequalities, and show that they define facets of this polytope. This answers a question posed by Qi and Sun (2000). Moreover, we show that we can separate these inequalities in polynomial time. Finally, we assess the computational relevance of the new inequalities by performing (limited) computational experiments.

1. Introduction and motivation

The axial 3-index (or 3-dimensional) assignment problem (3AP) can be described as follows. Given are three disjoint n-sets I, J, K and a weight function $w : I \times J \times K \longrightarrow \mathbb{R}$. The problem is to select a collection of triples $M \subseteq I \times J \times K$ such that each element of each set appears in exactly one triple, and such that total weight of the selected triples is minimized (or maximized). Its formulation as an Integer Linear Program (ILP) is:

$$\min \sum_{i\in I}\sum_{j\in J}\sum_{k\in K} w_{ijk}x_{ijk}$$

$$\text{s.t.} \quad \sum_{j\in J}\sum_{k\in K} x_{ijk} = 1 \quad \forall i \in I, \tag{1.1}$$

$$\sum_{i\in I}\sum_{k\in K} x_{ijk} = 1 \quad \forall j \in J, \tag{1.2}$$

$$\sum_{i\in I}\sum_{j\in J} x_{ijk} = 1 \quad \forall k \in K, \tag{1.3}$$

$$x_{ijk} \in \{0, 1\} \quad \forall i \in I, j \in J, k \in K. \tag{1.4}$$

The 3AP is a straightforward generalization of the well-known (two-dimensional) assignment problem. Whereas the latter problem is solvable by a polynomial-time algorithm, the 3AP is more difficult: no polynomial-time algorithm is known for the 3AP, see [17]. The 3AP however, is a very relevant problem, and has applications in many different fields of science. In fact, the above stated formulation can be found in recent papers that deal with the statistical design of experiments. For instance, Rassen et al. [24], Higgins [14], and Xu and Kalbfleisch [26] describe how subjects, each receiving one of three

* Corresponding author.
E-mail addresses: t.dokka@lancaster.ac.uk (T. Dokka), frits.spieksma@kuleuven.be (F.C.R. Spieksma).

http://dx.doi.org/10.1016/j.dam.2015.07.018

possible treatments, should be assembled into triples in a best possible way. A completely different application can be found in the field of computational chemistry where so-called methyl groups need to be assigned to minimize the cost of the resulting resonance assignment; we refer to John et al. [16] for further details. Yet another application is described in computational biology (see Biyani et al. [7]).

Another reason for the importance of the 3AP is that it can be seen as a special case of the axial multi-index assignment problem (mAP). In this case, instead of three disjoint n-sets, we are given m disjoint n-sets, and the problem is to find nm-tuples such that each element is in exactly one m-tuple, while minimizing total cost. This problem is particularly relevant in target tracking situations, which occur not only in data-association (see e.g. Poore and Gadaleta [20] and the references contained therein), but also in particle tracking in live-cell imaging studies, see Feng et al. [12] for an example.

A consequence of these different applications is the existence of a wide range of heuristic solution methods for the 3AP. Many of the papers above, as well as Huang and Lim [15] and Aiex et al. [1] describe heuristic procedures. And although our work reported here is not primarily algorithmic in nature, we remark that the inequalities described here can be used in an (exact) cutting-plane approach, and hence can also be used to establish lower bounds (see Section 5), thereby helping to assess the quality of heuristic solutions found.

Thus, in this work we contribute to the polyhedral knowledge of the facial structure of the convex hull of the feasible solutions to (1.1)–(1.4). First, we describe known classes of facets by adopting a geometrical point of view, i.e., we organize the variables x_{ijk} in a three-dimensional array (a cube). This allows us to illustrate the differences between distinct classes of inequalities (Section 2). Next, we give a new class of facet-defining inequalities, called the wall inequalities (Section 3). We show that this class can be separated in polynomial time in Section 4. Further, we perform limited computational experiments in order to assess the practical relevance of the wall inequalities in Section 5.

1.1. Literature

It is well-known that, as opposed to the polytope that corresponds to the two-dimensional assignment problem, not all extreme vertices of the polytope corresponding to (1.1)–(1.4) are integral. In fact, different types of fractional vertices exist; work on this topic is reported in Kravtsov [18]. Early work investigating the facial structure of the polytope P_I is described in Balas and Saltzman [5] and Euler [10]. They give different classes of facet-defining inequalities (see Section 2). Subsequently, other classes of facet-defining inequalities are reported in Qi and Balas [21] (see also Qi, Balas and Gwan [22]). Separation algorithms are discussed in Balas and Qi [4]. A nice overview of existing polyhedral results is given in Qi and Sun [23]. This paper also contains the question: "Are there other facet classes such that the right hand sides of their defining inequalities are 2?", to which we provide an (affirmative) answer here. An exact algorithm based on known valid inequalities that are used in conjunction with Lagrangian multipliers is given in Balas and Saltzman [6].

A related polytope is the one that corresponds to the so-called *planar* three-index assignment problem; this is the problem that arises when a collection of triples needs to be selected such that each *pair* of elements from $(I \times J) \cup (I \times K) \cup (J \times K)$ is selected precisely once. The facial structure of this polytope has first been studied in Euler et al. [11]. Also, polytopes that correspond to four-index assignment problems have been studied, see Appa et al. [2]. Recent results that unify these polyhedral results for all multi-index assignment polytopes can be found in Appa et al. [3]. We refer to [25] for results concerning approximability of special cases of 3AP.

1.2. Preliminaries

To avoid trivialities we assume $n \geq 4$. Let A^n denote the (0, 1) matrix corresponding to the constraints (1.1)–(1.3). Thus A^n has n^3 columns (one for each variable) and $3n$ rows (one for each constraint). Then, the 3-index assignment polytope is the following object:

$$P_I^n = \text{conv}\{x \in \{0, 1\}^{n^3} : A^n x = \mathbf{1}\},$$

while its linear programming (LP) relaxation is described as:

$$P^n = \{x \in R^{n^3} : A^n x = \mathbf{1}, x \geq 0\}.$$

For reasons of convenience, we will often omit the superscript n, and use A, P_I and P instead. We use $R \equiv (I \cup J \cup K)$; elements of R are called *indices*. We also use $V \equiv I \times J \times K$; elements of V are called *triples*. Given a triple $(i, j, k) \in V$, we refer to i, j and k as *first*, *second*, and *third* indices respectively.

An important object is the so-called *column intersection* graph corresponding to A^n. This graph $G(V, E)$, has a node for each column of A^n (i.e., a node for each triple) and an edge for every pair of columns that have a $+1$ entry in the same row. Notice that each column of A^n contains three $+1$'s. The support of the intersection of two columns c and d is nothing else but the number of indices that the triples c and d have in common; this number is denoted by $|c \cap d|$. Thus, the edge set E of the column intersection graph is given by $E = \{(c, d) : \{c, d\} \subseteq V, |c \cap d| \geq 1\}$, i.e., two nodes are connected iff the corresponding triples share some index. We call two triples *disjoint* if the corresponding nodes are not connected in G. Clearly, *cliques* (a complete subgraph of G) and *odd cycles* (a cycle consisting of an odd number of vertices in G) are relevant structures. Indeed, it is clear that when given a set of variables that correspond to nodes that form a clique in G, at most one