European Journal of Operational Research 247 (2015) 745–754

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

The Partitioning Min–Max Weighted Matching Problem

Dominik Kress, Sebastian Meiswinkel*, Erwin Pesch

Department of Management Information Science, University of Siegen, Kohlbettstr. 15, D-57068 Siegen, Germany

ARTICLE INFO

Article history:
Received 11 July 2014
Accepted 16 June 2015
Available online 20 June 2015

Keywords:
Assignment
Partitioning
Maximum matching
Bipartite graph
Container transshipment

ABSTRACT

We introduce and analyze the Partitioning Min–Max Weighted Matching (PMMWM) Problem. PMMWM combines the problem of partitioning a set of vertices of a bipartite graph into disjoint subsets of restricted size and the strongly NP-hard Min–Max Weighted Matching (MMWM) Problem, that has recently been introduced in the literature. In contrast to PMMWM, the latter problem assumes the partitioning to be given. Applications arise in the field of intermodal container terminals and sea ports. We propose a MILP formulation for PMMWM and prove that the problem is NP-hard in the strong sense. Two heuristic frameworks are presented. Both of them outperform standard optimization software. Our extensive computational study proves that the algorithms provide high quality solutions within reasonable time.

1. Introduction

In this paper we consider a variant of the strongly NP-hard Min–Max Weighted Matching (MMWM) Problem, that has recently been introduced by Barketau, Pesch, and Shafransky (2015). An instance of MMWM is defined by an edge-weighted bipartite graph $G(U, V, E)$ with disjoint vertex sets U and V (bipartitions), edge set E, and a partitioning of U into disjoint subsets (components). Given a maximum matching on G, the weight of a component is defined as the sum of the weights of the edges of the matching that are incident to the vertices of the component. The objective is to find a maximum matching that minimizes the maximum weight of the components. The components may, for example, correspond to areas of responsibility of managers or tasks to be performed by a worker or machine. The objective is to balance the workload, risk, etc. over these components.

While Barketau et al. (2015) assume the components to be fixed, we relax this assumption by assuming the partitioning decision to be part of the optimization, with only the desired number of components being fixed. We refer to this problem as the Partitioning Min–Max Weighted Matching (PMMWM) Problem. Fig. 1 illustrates an exemplary solution to an example instance of PMMWM. The maximum matching is represented by bold edges. Edge weights are solely depicted for the edges of the matching. Bipartition $U = \{u_1, \dots, u_7\}$ has been partitioned into the components U_1, U_2 and U_3 with weights 6, 9 and 4, respectively. Hence, the corresponding objective function value of the PMMWM instance is $\max\{6, 9, 4\} = 9$. If we move u_4 to U_3 without changing the matching, the objective function value reduces by 1.

This paper is organized as follows. In Section 2 we provide a detailed problem description along with a MILP model of the problem. We present two applications of PMMWM in the context of a reach stacker based container terminal and a rail-road terminal. Next, a proof of the problem's strong NP-hardness is given in Section 3. Section 4 introduces two heuristic frameworks that are being analyzed based on computational tests in Section 5. In Section 6, we summarize the findings of this paper.

2. Detailed problem definition and applications

Let $G(U, V, E)$ be a weighted bipartite graph with bipartitions U and V and edge set E. The elements of U and V are indexed $i = 1, \dots, n_1$ and $j = 1, \dots, n_2$, respectively. Assume $n_1 \le n_2$. A weight $c(e) = c_{uv} \in \mathbb{Q}_0^+$ is associated with each edge $e = (u, v) \in E$ of G. Define a matching as a set $M \subseteq E$ of pairwise nonadjacent edges and a maximum matching as a matching having the largest possible size $|M|$ amongst all matchings on G. Throughout the paper, we will assume that, for any given bipartite graph, there exists a maximum matching Π with $|\Pi| = n_1$. As in Barketau et al. (2015), given a partitioning of U into m disjoint subsets, $U_1, U_2, \dots, U_m$, the value of a maximum matching Π is defined to be $w(\Pi) := \max_{k\in\{1,\dots,m\}}\{\sum_{u\in U_k,(u,v)\in\Pi} c_{uv}\}$ (refer to Fig. 1 for an illustration). Then PMMWM can formally be defined as follows: find a partitioning of the vertex set U into m (potentially empty) disjoint subsets, $U_1, U_2, \dots, U_m$, with at most $\bar{u}$ elements in each subset, and a maximum matching Π on G, such that the value

* Corresponding author. Tel.: +49 271 740 4597; fax: +49 271 740 2940.
E-mail addresses: dominik.kress@uni-siegen.de (D. Kress), sebastian.meiswinkel@uni-siegen.de (S. Meiswinkel), erwin.pesch@uni-siegen.de (E. Pesch).

http://dx.doi.org/10.1016/j.ejor.2015.06.041
0377-2217/

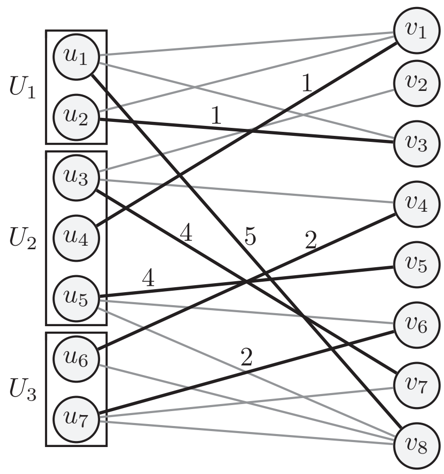

Fig. 1. A solution to an example instance of PMMWM.

of Π is minimum amongst all maximum matchings over all possible partitionings of U.

We define the following binary variables:

$$x_{ij} := \begin{cases} 1 & \text{if } (i,j) \in \Pi, \\ 0 & \text{else}, \end{cases} \quad \forall\, (i,j) \in E \tag{1}$$

and

$$y_{ik} := \begin{cases} 1 & \text{if } i \in U_k, \\ 0 & \text{else}, \end{cases} \quad \forall\, i \in U, k \in \{1, \ldots, m\}. \tag{2}$$

Then a nonlinear mathematical model for PMMWM is as follows:

$$\min_{\mathbf{x},\mathbf{y}} \quad \max_{k \in \{1,\ldots,m\}} \left\{ \sum_{i \in U} \sum_{j \in V} c_{ij} y_{ik} x_{ij} \right\} \tag{3}$$

$$\text{s.t.} \quad \sum_{j \in V} x_{ij} = 1 \quad \forall\, i \in U, \tag{4}$$

$$\sum_{i \in U} x_{ij} \le 1 \quad \forall\, j \in V, \tag{5}$$

$$\sum_{k=1}^{m} y_{ik} = 1 \quad \forall\, i \in U, \tag{6}$$

$$\sum_{i \in U} y_{ik} \le \bar{u} \quad \forall\, k \in \{1, \ldots, m\}, \tag{7}$$

$$x_{ij} \in \{0, 1\} \quad \forall\, (i,j) \in E, \tag{8}$$

$$y_{ik} \in \{0, 1\} \quad \forall\, i \in U,\ k \in \{1, \ldots, m\}. \tag{9}$$

The objective function (3) minimizes the value of the maximum matching over all possible partitionings and all maximum matchings. Constraints (4)–(5) are well known maximum matching constraints (recall that $n_1 \le n_2$). Constraints (6) enforce every vertex $u \in U$ to be an element of exactly one partition U_k, $k \in 1, \ldots, m$. Constraints (7) restrict the number of vertices in each partition to be at most $\bar{u}$. The domains of the variables are defined by (8)–(9).

A specific application of PMMWM arises at small to medium sized sea ports where containers are handled by reach stackers. The corresponding terminals, as schematically represented in Fig. 2, can include large long-term storage areas and additional temporary storage areas (or marshaling-areas; see, for instance, Preston & Kozan, 2001; Kozan & Preston, 1999). The latter areas aim at improving the performance of the terminals by inducing short turnaround times of vessels when distances to long-term storage areas are relatively large. When a vessel arrives at a berth at the terminal, containers are unloaded by quay cranes and then stored in a temporary storage area that is located next to the berth. Containers that leave the terminal by ship are moved to the temporary storage area using reach stackers during previous idle times. We will assume that these vehicles are "fast" if they are unloaded and "slow" if they are loaded and can thus restrict ourselves to considering the movements of loaded vehicles only. This is a common assumption when considering container movements (see, for instance, Boysen & Fliedner, 2010) and is supported by the fact, that "reach stackers [in comparison to straddle carriers] are less stable in the forward direction as the machines will fall forward when breaking in an emergency, particularly if the load is carried high for visibility reasons" (Isoloader, 2012). Then an application of PMMWM arises, when considering the process of emptying or refilling the

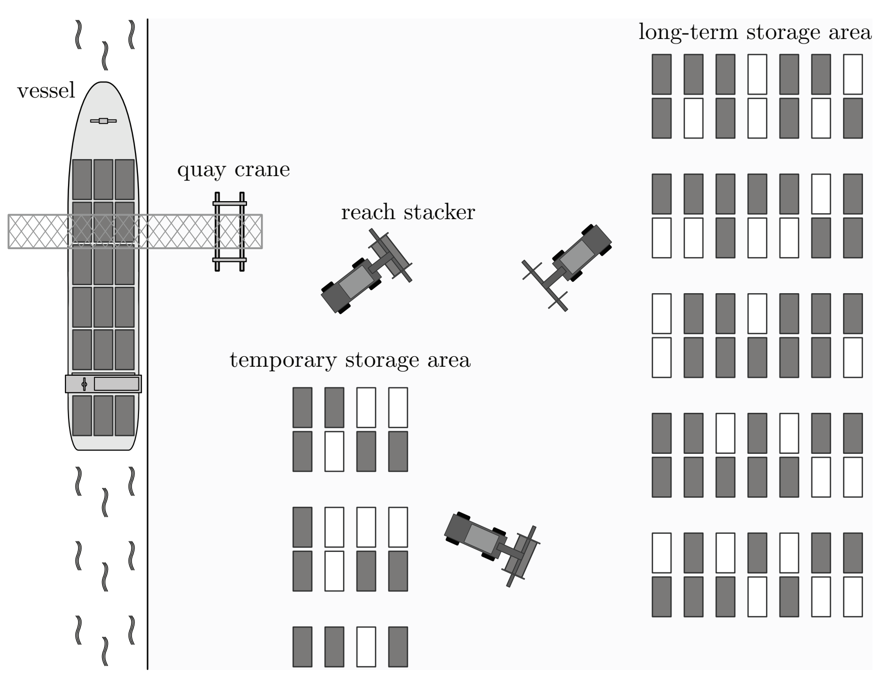

Fig. 2. Potential schematic layout of a reach stacker based terminal.