NETWORK TRANSFORMATION HEURISTICS FOR MULTI-STORY STORAGE RACK PROBLEMS

Yoshiyuki KARUNO
Department of Mechanical and System Engineering
Kyoto Institute of Technology
Matsugasaki, Sakyō-ku, Kyoto-shi, Kyoto 606-8585, Japan
karuno@kit.ac.jp

Hiroshi NAGAMOCHI
Department of Applied Mathematics and Physics
Kyoto University
Yoshida-honmachi, Sakyō-ku, Kyoto-shi, Kyoto 606-8501, Japan
nag@amp.i.kyoto-u.ac.jp

Abstract

In this paper, we consider a combinatorial optimization problem of allocating n items in a storage rack with multiple stories, where every story has linearly aligned m identical slots. Each item $J_i$ is characterized by three integers, an arrival time $a_i$, a storage duration $p_i$, and the number $q_i (\leq m)$ of consecutive slots required by the item. The storage rack has to start the storage service to an item $J_i$ promptly after it arrives, and has to maintain the item exactly until time $t = a_i + p_i$, selecting $q_i$ consecutive slots of a certain story. Each slot can maintain at most one item at a time, and no preemption of the storage service is allowed. The objective is to find a feasible allocation of the $n$ items that minimizes the number of dispatched stories of the storage rack. For the NP-hard allocation problem, we propose a polynomial time heuristic algorithm based on a transformation into a bipartite maximum matching problem, and prove that the approximation ratio is at most $\alpha + 1/z^*$, where $\alpha = \lfloor m/q_{\min} \rfloor / \lfloor m/q_{\max} \rfloor$, $q_{\min} = \min_{1 \leq j \leq n} (q_j)$, $q_{\max} = \max_{1 \leq j \leq n} (q_j)$, and $z^* (\geq 1)$ is the minimum of the number of dispatched stories.

Keywords: Combinatorial optimization, storage allocation, heuristic algorithms, maximum matching, performance guarantee.

1. INTRODUCTION

In this paper, we consider an allocation problem of n items in a storage rack with multiple stories, where every story has linearly aligned m identical slots. Each item $J_i$ is characterized by a 3-tuple $(a_i, p_i, q_i)$ of non-negative integers, where $a_i$ is the arrival time, $p_i$ is the storage duration (i.e., the service time) of item $J_i$, and $q_i$ is the number of consecutive slots of a certain story required by item $J_i$ (from which, we can represent each item $J_i$ by a rectangle with the size of $p_i \times q_i$ in a geometrical interpretation). The set of consecutive slots required by an item is not prescribed, but is selected. The storage rack has to start the storage service to an item $J_i$ promptly after it arrives (i.e., the storage rack has to start the storage service exactly at time $t = a_i$), and has to maintain the item exactly until time $t = a_i + p_i$, selecting $q_i$ consecutive slots of a certain story (and hence, no item can extend its occupation to more than one stories). Each slot can maintain at most one item at a time (i.e., no pair of items can employ a common slot at a time), and no preemption of the storage service is allowed. We refer to the restriction of starting times of the storage services as prompt service constraint, and the consequentness of selecting slots of a certain story as alignment constraint. The objective is to find a feasible allocation of the $n$ items that minimizes the number of dispatched stories of the storage rack. We call the problem MSR (Multi-story Storage Rack) for short. We here emphasize that all the 3-tuples of $(a_i, p_i, q_i)$ are given in the modeling. That is, problem MSR is not of on-line setting, but of off-line setting.

A decision version of problem MSR asks whether all $n$ items can be served by using linearly aligned $m$ slots of a single story or not. It has already been known that the decision version is NP-hard for an arbitrary $m$ (e.g., the full version of Buchbaum et al. [2], Garey and Johnson [4]). It is natural to ask how many seamless slots (by which we mean all dispatched slots are assumed to be situated on a single story) are required for serving all $n$ items, instead of asking the number of dispatched stories to be required. Such a problem has been studied extensively in computer science, under the name of dynamic storage allocation (DSA for short). Of course, problem DSA is NP-hard [2, 4]. Gergov proposed a 5-approximation algorithm to problem DSA [5], and afterward claimed a 3-approximation algorithm [6]. A $(2 + e)$-approximation algorithm for problem DSA was presented by Buchbaum et al. [1]. For the multi-device Balancing DSA, we are given $k$ devices (in this paper, we call them stories, and the number of stories is not given but is...
a decision variable), and the objective is to find a feasible allocation of the items which minimizes the maximum of the number of dispatched slots over the \( k \) devices (i.e., allocating the items to the \( k \) devices in a balanced manner). Wu et al. treated the multi-device Balancing DSA of on-line setting [10].

In this paper, we propose a polynomial time heur- istic algorithm to problem MSR, where the problem is approximately transformed into a bipartite maximum matching problem. It is well-known that the bipartite maximum matching problem can be solved in polynomial time (see Hopcroft and Karp [7], see also Korte and Vygen [9]). We also prove that the approximation ratio is at most \( \alpha + 1/z^* \), where \( \alpha = [m/q_{\text{min}}]/[m/q_{\text{max}}], q_{\text{min}} = \min_{1 \leq j \leq n} (q_j), q_{\text{max}} = \max_{1 \leq j \leq n} (q_j) \), and \( z^* \) is the minimum of the number of dispatched stories.

2. PROBLEM DESCRIPTION

We are given a set \( \mathcal{J} = \{ J_j \mid j = 1, 2, \ldots, n \} \) of \( n \) items to be served by a storage rack with multiple stories. Every story consists of linearly aligned \( m \) identical slots. Let \( z \) be a positive integer by which we denote the number of stories of the storage rack. Note that the \( z \) is not a given integer, but it is a decision variable in problem MSR. We denote the slots in all stories by \( S_1, S_2, \ldots, S_{5m} \), and a subset \( \{ S_i \} \) of \( h - i + 1 \) slots with consecutive indices by \( Q(i, h) \) \((1 \leq i \leq h \leq z \times m) \). In this paper, we assume that the \( m \) slots on each story are linearly aligned in the order of their indices, and hence the set of \( m \) slots on the \( k \)-th story is given by

\[
S_k = Q((k - 1) \times m + 1, k \times m).
\]

The definition implies \( Q(1, z \times m) = \bigcup_{k=1}^z S_k \). Each slot can maintain at most one item at a time.

Each item \( J_j \) is characterized by a 3-tuple \((a_j, p_j, q_j)\) of integers, where

- \( a_j (\geq 0) \): arrival time of item \( J_j \),
- \( p_j (> 0) \): storage duration of item \( J_j \),
- \( q_j (1 \leq q_j \leq m) \): the number of consecutive slots of a certain story required by item \( J_j \).

We denote by \( I = (m, n, a, p, q) \) an instance of problem MSR. For a given instance \( I \) of problem MSR, an allocation can be represented by a mapping \( \sigma : \mathcal{J} \rightarrow \mathbb{N} \), such that for an item \( J_j \), the \( q_j \) consecutive slots of \( Q(\sigma(J_j), \sigma(J_j) + q_j - 1) \) are selected during the storage period \([a_j, a_j + p_j]\), where \( \sigma(J_j) \) denotes the index of its tip slot.

An allocation \( \sigma \) is feasible if it satisfies, for any item \( J_j \),

\[
Q(\sigma(J_j), \sigma(J_j) + q_j - 1) \subseteq S_k \quad \text{for} \quad k \in \{1, 2, \ldots, z\}, \tag{1}
\]

and for any pair of items \( J_j \) and \( J_h \) \((j \neq h)\), either

\[
[a_j, a_j + p_j] \cap [a_h, a_h + p_h] = \emptyset \quad \text{or} \quad (2)
\]

\[
Q(\sigma(J_j), \sigma(J_j) + q_j - 1) \cap Q(\sigma(J_h), \sigma(J_h) + q_h - 1) = \emptyset. \tag{3}
\]

The objective is to find a feasible allocation \( \sigma = \sigma^* \) that minimizes the number \( z(\sigma) \) of dispatched stories of the storage rack. We refer to the \( \sigma^* \) as an optimal allocation, and the minimum \( z^* = z(\sigma^*) \) of the number of dispatched stories as the optimal value.

Example 1 We are given eight items \((\text{i.e., } n = 8)\) to be served by a storage rack where every story has three slots \((\text{i.e., } m = 3)\). The items are characterized by

- \((a_1, p_1, q_1) = (0, 4, 2)\),
- \((a_2, p_2, q_2) = (1, 2, 1)\),
- \((a_3, p_3, q_3) = (2, 7, 1)\),
- \((a_4, p_4, q_4) = (3, 2, 2)\),
- \((a_5, p_5, q_5) = (4, 3, 2)\),
- \((a_6, p_6, q_6) = (5, 3, 1)\),
- \((a_7, p_7, q_7) = (7, 2, 3)\),
- \((a_8, p_8, q_8) = (8, 2, 1)\).

Figure 1 illustrates the following feasible allocation \( \sigma \) for the above instance of problem MSR.

\[
\sigma[J_1] = 1, \quad \sigma[J_2] = 3, \quad \sigma[J_3] = 6, \quad \sigma[J_4] = 4, \quad \sigma[J_5] = 1, \quad \sigma[J_6] = 4, \quad \sigma[J_7] = 1, \quad \sigma[J_8] = 4.
\]

The first story consists of slots \( S_1, S_3, \) and \( S_5 \), and the second story consists of slots \( S_4, S_5, \) and \( S_6 \). Four items \( J_1, J_3, J_5, \) and \( J_7 \) are allocated on the first story, while the remaining four items \( J_3, J_4, J_6, \) and \( J_8 \) are allocated on the second story, i.e., \( z(\sigma) = 2 \). Note that no item extends its occupation to more than one stories (for example, if we change the tip slot \( \sigma[J_4] = 4 \) of item \( J_4 \) only in the \( \sigma \) by \( \sigma'[J_4] = 3 \) to obtain a new allocation \( \sigma' \), then the \( \sigma' \) is not feasible) (see Eq. (1)). By regarding the time period of \([3, 4]\), since \( q_1 + q_3 + q_4 = 5 \) and \( m = 3 \), a straightforward lower bound on the optimal value is obtained as \( z^* \geq \lceil (q_1 + q_3 + q_4)/m \rceil = 2 \). Hence, an optimal allocation is the feasible allocation \( \sigma = \sigma^* \), and we have \( z^* = 2 \).
3. NETWORK TRANSFORMATION

In this section, we propose a polynomial time heuristic algorithm to problem MSR based on an approximate transformation into a bipartite maximum matching problem. Let

\[ q_{\text{min}} = \min \{ q_j \mid J_j \in \mathcal{J} \}, \]

and let

\[ q_{\text{max}} = \max \{ q_j \mid J_j \in \mathcal{J} \}. \]

We will prove the following theorem.

**Theorem 1** Problem MSR is \((\alpha + 1/z^*)\)-approximable in \(O(n^{2.5})\) time, where \(\alpha = \lfloor m/q_{\text{min}} \rfloor / \lfloor m/q_{\text{max}} \rfloor\) and \(z^*\) is the optimal value for a given instance with \(n\) items.

Precisely speaking, for a given instance \(I = (m, n, a, p, q)\) of problem MSR, a feasible allocation \(\mathcal{S}\) obtained by the proposed heuristic, which runs in \(O(n^{2.5})\) time, satisfies

\[ \frac{z(\mathcal{S})}{z^*} \leq \frac{\lfloor m/q_{\text{min}} \rfloor}{\lfloor m/q_{\text{max}} \rfloor} + \frac{1}{z^*}. \quad (4) \]

3.1 Lower Bound via Maximum Matching

In this subsection, we compute a lower bound on the number of slots required to serve all \(n\) items in a given instance \(I = (m, n, a, p, q)\) of problem MSR. For this, we regard the number \(q_j\) of consecutive slots required by each item \(J_j\) as 1 so that the alignment constraint is no longer effective, and consider an allocation of such items that requires the minimum number \(k_{\text{min}}\) of slots. In other words, we wish to find the maximum number \(\mu\) of items \(J_j\) that are followed by some other items \(J_k\) in a slot, since \(k_{\text{min}} = n - \mu\) holds. Such a solution can be identified as a maximum matching in a bipartite graph \(G_I = (R_I \cup S_I, E_I)\), which is constructed associated with the data \((n, a, p)\) of instance \(I\) as follows: Let the two vertex sets be given by \(R_I = \{ r_j \mid J_j \in \mathcal{J} \}\) and \(S_I = \{ s_j \mid J_j \in \mathcal{J} \}\), and let the edge set \(E_I\) consist of edges \((r_j, s_k)\) for all pairs of two distinct items \(J_j, J_k\) in \(I\) such that

\[ a_j + p_j \leq a_k \quad (5) \]

holds (see Eq. (2)).

Observe that the bipartite graph \(G_I\) has a matching \(M \subseteq E_I\) if and only if all \(n\) items \(J_j\) with \((a_j, p_j, 1)\) can be served using \(n - |M|\) slots.

A maximum matching in a graph \(G = (V, E)\) can be obtained in \(O(|V|^2)\) time (see Hopcroft and Karp [7]). Since \(G_I\) contains \(2n\) vertices, we can compute a maximum matching \(M\) in \(G_I\) and \(k_{\text{min}} = n - \mu = n - |M|\) in \(O(n^{2.5})\) time (we can also refer to an interval-graph coloring problem, e.g., see Cormen et al. [3], instead of the maximum matching).

**Lemma 1** For an instance \(I\) of problem MSR, let \(z^*\) be the optimal value. Then it satisfies

\[ z^* \geq \frac{k_{\text{min}}}{\lfloor m/q_{\text{min}} \rfloor}. \quad (6) \]

**Proof:** To derive a contradiction, we indirectly assume that \(z^* < k_{\text{min}} / \lfloor m/q_{\text{min}} \rfloor\). Let \(J^{(k)}\) be the set of items assigned to the \(k\)-th story \((k = 1, 2, \ldots, z^*)\) in an optimal allocation, and \(k^{(k)}\) be the maximum number of items in \(J^{(k)}\) such that their durations overlap each other in the optimal allocation at a certain time, where \(k^{(k)} \leq \lfloor m/q_{\text{min}} \rfloor\) holds. Now reduce the number \(q_j\) of consecutive slots required by each item \(J_j\) to 1. Then we easily observe that all the items in \(J^{(k)}\) with the reduced values can be served using \(k^{(k)}\) slots. Hence the instance \(I\) has \(\sum_{k \leq k^{(k)}} k^{(k)}\) sequences of items such that two consecutive items \(J_j\) and \(J_k\) in each sequence satisfy \(a_j + p_j \leq a_k\). However, \(\sum_{k \leq k^{(k)}} k^{(k)} \leq z^*/\lfloor m/q_{\text{min}} \rfloor < k_{\text{min}}\) contradicts that \(k_{\text{min}}\) is the minimum number of such sequences of \(J\).

3.2 Proposed Heuristic

Given a maximum matching \(M \subseteq E_I\), we have \(k_{\text{min}}\) sequences \(P_1, P_2, \ldots, P_{\text{num}}\) of items such that an item \(J_j\) is immediately followed by another item \(J_k\) if \((r_j, s_k) \in M\).

Next, based on the set of \(k_{\text{min}}\) sequences \(P_1, P_2, \ldots, P_{\text{num}}\) of items, the proposed heuristic prepares \(k_{\text{min}}\) disjoint sub-sets of slots, each of which has \(q_{\text{max}}\) consecutive slots of a certain story. We simply call the \(\ell\)-th subset of \(q_{\text{max}}\) consecutive slots the \(\ell\)-th slot group, \(\ell = 1, 2, \ldots, k_{\text{min}}\).

For any \((r_j, s_k) \in M\), any slot group with \(q_{\text{max}}\) consecutive slots can start the storage service to item \(J_k\) at its arrival time \(t = a_k\) after completing item \(J_j\) exactly at time \(t = a_j + p_j\) (see Eq. (5) again). Thus, all items in a sequence \(P_t\) can be served by any slot group exactly at their arrival times.

The proposed heuristic finally assigns a set of \([m/q_{\text{max}}]\) slot groups (e.g., from the first to the \([m/q_{\text{max}}]\)-th ones) into a story, and a different set of \([m/q_{\text{max}}]\) slot groups (e.g., from \((m/q_{\text{max}}) + 1\)-th to \((2 \times [m/q_{\text{max}}])\)-th ones) into another story, and so on. Hence \([k_{\text{min}}/\lfloor m/q_{\text{max}} \rfloor\) stories are dispatched in the storage rack. We denote by \(\mathcal{S}\) a feasible allocation obtained in such a manner. If \(k_{\text{min}} \leq \lfloor m/q_{\text{max}} \rfloor\), then the proposed heuristic can allocate all \(n\) items into a single story, i.e., \(z(\mathcal{S}) = 1\), and the \(\mathcal{S}\) is an optimal allocation. Hence, we can treat only the cases of \(k_{\text{min}} > \lfloor m/q_{\text{max}} \rfloor\).

The remaining \((m \mod q_{\text{max}})\) slots of each story are not going to be used in the performance analysis of this paper (but we can include these slots in a certain implementation of the proposed heuristic since it may be more advantageous for the empirical performance).

The time complexity of the proposed heuristic can be concluded to be \(O(n^{2.5})\), since the part of solving the maximum matching problem requires the largest time complexity.

3.3 Performance Analysis

We remind the readers of the number

\[ z(\mathcal{S}) = \left[ \frac{k_{\text{min}}}{\lfloor m/q_{\text{max}} \rfloor} \right] \quad (7) \]
of dispatched stories obtained by the proposed heuristic. From this, we have

\[
z(\hat{\sigma}) = \left[ \frac{\kappa_{\min}}{m/q_{\max}} \right] \quad \text{(Eq. (7) again)}
\]

\[
\leq \frac{\kappa_{\min}}{m/q_{\max}} + 1
\]

\[
= \frac{\kappa_{\min} \times m/q_{\min}}{m/q_{\max} \times m/q_{\min}} + 1
\]

\[
\leq \frac{m/q_{\min}}{m/q_{\max}} \times z' + 1 \quad \text{(by Eq. (6))}
\]

which completes the proof of Theorem 1.

In closing this section, we additionally make the following remark. Suppose that the storage rack with a single story is allowed to possess a number of seamless slots, instead of the limited number \( m \). Then, the proposed heuristic dispatches \( \kappa_{\min} \times q_{\max} \) slots, while an optimal allocation dispatches at least \( \kappa_{\min} \times q_{\min} \) slots (see the proof of Lemma 1). Hence, the proposed heuristic delivers a \( q_{\max}/q_{\min} \)-approximate allocation to problem DSA. This performance guarantee implies a duality of the network transformation to that based on a minimum cost flow problem for a variant of problem DSA (e.g., see Karuno et al. [8]), which asks to find a feasible allocation that maximizes the weighted number of accepted items by a storage rack only with a single story having the limited number \( m \) of linearly aligned identical slots.

4. CONCLUDING REMARKS

In this paper, we considered a combinatorial optimization problem of allocating \( n \) items in the storage rack with multiple stories, where every story has linearly aligned \( m \) identical slots, which we called problem MSR (Multi-story Storage Rack) for short. An integral 3-tuple \((a_j, p_j, q_j)\) characterizes each item \( J_j \), where \( a_j \) is the arrival time, \( p_j \) is the storage duration, and \( q_j \) is the number of consecutive slots of a certain story required by the item. The objective is to find a feasible allocation of the \( n \) items that minimizes the number of dispatched stories of the storage rack. In this paper, we proposed an \( O(n^{2.5}) \) time heuristic algorithm based on a transformation into a maximum matching problem, and proved that the approximation ratio is at most \( \alpha + 1/z' \), where \( \alpha = [m/q_{\min}]/[m/q_{\max}] \) and \( z' \) is the minimum of the number of dispatched stories of the storage rack for a given instance. In addition, the proposed heuristic can be modified so that it delivers a \( q_{\max}/q_{\min} \)-approximate allocation to problem DSA.

The following directions for future research are open. It would be significant to investigate the existence of a constant factor approximation algorithm for problem MSR. It would be interesting to make some procedure for improving the empirical performance of the proposed heuristic. It would also be interesting to ask how many number of stories are reduced if the storage rack is allowed to give the storage service with preemption to the items.

ACKNOWLEDGMENT

We would like to express our gratitude to the anonymous referees whose comments include some information of the multi-device Balancing DSA (i.e., Reference [10]). This research was partially supported by a Grant-in-Aid for Scientific Research (C) of the Japan Society for the Promotion of Science (JSPS).

REFERENCES


