On the Bandpass problem

Guohui Lin

Department of Computing Science, University of Alberta
Edmonton, Alberta T6G 2E8, Canada.

October 1, 2009

Abstract

The complexity of the Bandpass problem is re-investigated. Specifically, we show that the problem with any fixed bandpass number $B \geq 2$ is NP-hard. Next, a row stacking algorithm is proposed for the problem with three columns, which produces a solution that is at most 1 less than the optimum. For the special case $B = 2$, the row stacking algorithm guarantees an optimal solution. On approximation, for the general problem, we present an $O(B^2)$-algorithm, which reduces to a 2-approximation algorithm for the special case $B = 2$.

Keywords: Bandpass problem, Hamiltonian path problem, NP-hard, approximation algorithm, exact algorithm

1 Introduction

The Bandpass problem was the first time formulated and presented in the Annual INFORMS meeting, October 2004, Denver, CO, USA, 2004 [3]. Some complexity and algorithmic results were recently summarized in [2], where a library of instances are included to challenge the algorithm design. The Bandpass problem can be easily described as follows. Given a binary matrix A of dimension $m \times n$, and a positive integer B called the *bandpass number*, a set of B consecutive non-zero elements in a column of the matrix is called a *bandpass*. Two bandpasses in the same column are not allowed to have common rows. The *Bandpass problem* is to find an optimal permutation of rows of the matrix such that the total number of extracted bandpasses is maximized.

This combinatorial optimization problem arises in optical communication networks, to design an optimal packing of information flows on different wavelengths into groups to obtain the highest available cost reduction using wavelength division multiplexing technology [2]. The matrix $A_{m \times n}$ represents a sending point which has m information packages to be sent to n different destination points, where $a_{ij} = 0$ if information package i is destined for point j, or $a_{ij} = 1$ otherwise. Roughly speaking, a bandpass provides an opportunity for merging information and thus to reduce the communication cost. Different bandpass numbers can be used in practice for cost reduction. However, according to [2], this would increase handling complexities and eventually induce additional costs. In the current version of the Bandpass problem, only one bandpass number is considered [2]. The interested readers might refer to [3, 2] for more details of the application.

Babayev *et al.* concluded that it is not reasonable to try to find an optimal solution to the Bandpass problem by exhaustive search over all row permutations [2]. They tried to study the computational complexity of the Bandpass problem, and modeled the problem as an integer program followed by calling existing integer programming solvers. Heuristics were also proposed to generate good solutions, though no guaranteed performance. Besides, they created a library of problem instances with known and unknown optimal solutions, open for public use.

Unfortunately, the proof of the NP-hardness for the Bandpass problem, via a reduction from 3SAT, provided in [5] does not seem correct. In fact, the presented reduction is from the Bandpass problem to 3SAT. Furthermore, the claim that *the Bandpass problem with three or more columns in the matrix is NP-hard* is probably wrong. In this paper, we prove the NP-hardness of the Bandpass problem with any fixed bandpass number $B \geq 2$ by providing a natural reduction from the Hamiltonian path problem [5]. For the problem containing exactly three columns, we present a row stacking algorithm which produces a solution at most 1 less than the optimum. Furthermore, such an algorithm gives an optimal solution for the special case $B = 2$. On approximation, an $O(B^2)$-algorithm is presented for the general problem, which reduces to a 2-approximation for the special case $B = 2$.

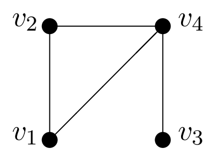
2 The complexity

The Bandpass problem looks for a permutation of rows of the given matrix to maximize the total number of bandpasses that can be extracted from the resultant matrix. In this section, we present a reduction from the Hamiltonian path problem [5] to prove the NP-completeness of the Bandpass decision problem. Given a graph, a Hamiltonian path in the graph is a permutation of vertices such that every two consecutive vertices in the permutation are adjacent (*i.e.*, connected by an edge) in the graph.

Hamiltonian Path (HP)
Instance: Graph $G = (V, E)$
Question: Does G contain a Hamiltonian path?

We construct an instance of the Bandpass decision problem as follows. Let vertices in V be labeled as $v_1, v_2, \ldots, v_n$. Each vertex v_i corresponds to three rows, indexed $3i-2, 3i-1, 3i$, in the target matrix A. The number of rows in A is $3n$ and the number of columns in A is $2n^2+\frac{1}{2}(n^2-n) = \frac{5}{2}n^2 - \frac{1}{2}n$. All entries in A are initialized to 0. Next we flip some entries to 1. Firstly, in the three rows corresponding to v_i, $a_{3i-2,j} = 1$ for $j = 2(i-1)n+1, 2(i-1)n+2, \ldots, 2(i-1)n+n$, $a_{3i-1,j} = 1$ for $j = 2(i-1)n+1, 2(i-1)n+2, \ldots, 2in$, and $a_{3i,j} = 1$ for $j = 2(i-1)n+n+1, 2(i-1)n+n+2, \ldots, 2in$. Columns indexed from $2n^2 + 1$ to $2n^2 + \frac{1}{2}(n^2 - n)$ in matrix A one-to-one correspond to pairs of vertices (v_{i_1}, v_{i_2}) where $i_1 < i_2$. For each edge $(v_{i_1}, v_{i_2}) \in E$, with $i_1 < i_2$, let j be the index of the corresponding column. Then, $a_{3i_1-2,j} = 1$, $a_{3i_1,j} = 1$, $a_{3i_2-2,j} = 1$, and $a_{3i_2,j} = 1$. Lastly, set $B = 2$, and the construction is complete. Given a sample graph showing in Figure 1(a), the matrix A constructed in the above way is illustrated in Figure 1(b).

In the following, we prove that graph G contains a Hamiltonian path if and only if there is a row permutation for matrix A which generates $2n^2 + (n-1)$ bandpasses. The "only if" part is fairly straightforward. Given a Hamiltonian path of graph G, without loss of generality, assume



(a)

	12345678	9–16	17–24	25–32	33	34	35	36	37	38
1	11110000	00000000	00000000	00000000	1	0	1	0	0	0
2	11111111	00000000	00000000	00000000	0	0	0	0	0	0
3	00001111	00000000	00000000	00000000	1	0	1	0	0	0
4	00000000	11110000	00000000	00000000	1	0	0	0	1	0
5	00000000	11111111	00000000	00000000	0	0	0	0	0	0
6	00000000	00001111	00000000	00000000	1	0	0	0	1	0
7	00000000	00000000	11110000	00000000	0	0	0	0	0	1
8	00000000	00000000	11111111	00000000	0	0	0	0	0	0
9	00000000	00000000	00001111	00000000	0	0	0	0	0	1
10	00000000	00000000	00000000	11110000	0	0	1	0	1	1
11	00000000	00000000	00000000	11111111	0	0	0	0	0	0
12	00000000	00000000	00000000	00001111	0	0	1	0	1	1

(b)

Figure 1: (a) A sample graph $G = (\{v_1, v_2, v_3, v_4\}, \{(v_1, v_2), (v_1, v_4), (v_2, v_4), (v_3, v_4)\})$, and (b) the constructed matrix A in the Bandpass instance, where columns 33–38 one-to-one corresponds to vertex pairs (v_1, v_2), (v_1, v_3), (v_1, v_4), (v_2, v_3), (v_2, v_4), and (v_3, v_4), respectively.

that the vertex order in the path is $v_1, v_2, \ldots, v_n$ (*i.e.*, $(v_i, v_{i+1}) \in E$, for $1 \le i \le n-1$). The matrix A constructed in the above way has one bandpass in each of the first $2n^2$ columns. Furthermore, one bandpass can be extracted from one among columns $2n^2+1$ to $2n^2+\frac{1}{2}(n^2-n)$ between rows $3i$ and $3(i+1)-2$, corresponding to vertices v_i and v_{i+1} respectively, for $i = 1, 2, \ldots, n-1$. This gives a total of $2n^2 + (n-1)$ bandpasses.

To prove the "if" part, we first notice that for rows indexed $3i-2$ and $3i$, the maximum possible number of bandpasses extracted from columns $2n^2+1$ to $2n^2+\frac{1}{2}(n^2-n)$ is at most $n-1$. The maximum case happens only if the degree of vertex v_i in graph G is $n-1$, and that these two rows are consecutive in the given row permutation. It follows that for at least one of these two rows, no bandpass can be extracted from columns 1 to $2n^2$. Moreover, for the row indexed $3i-1$, the maximum possible number of bandpasses extracted from all columns is reduced to n. Therefore, we can always move both rows $3i-2$ and $3i$ to line up with row $3i-1$, with an increasing number of bandpasses. This way, we can always, if necessary, obtain from the given row permutation another row permutation, in which rows indexed $3i-2, 3i-1, 3i$ appear sequentially. In this new row permutation, the number of bandpasses generated from columns 1 to $2n^2$ is exactly $2n^2$. Let the order of vertices corresponding to this row permutation be $v_{i_1}, v_{i_2}, \ldots, v_{i_n}$. It follows that for every $j = 1, 2, \ldots, n-1$, there is one bandpass extracted in between rows $3i_j$ and $3i_{j+1}-2$, from some column indexed $2n^2+1$ to $2n^2+\frac{1}{2}(n^2-n)$. Equivalently speaking, vertices v_{i_j} and $v_{i_{j+1}}$ are adjacent in graph G. That is, graph G contains a Hamiltonian path $\langle v_{i_1}, v_{i_2}, \ldots, v_{i_n} \rangle$.

For the matrix A illustrated in Figure 1(b), the number of bandpasses is 34. If swapping row group $\{7, 8, 9\}$ with row group $\{10, 11, 12\}$, while maintaining their internal sequential orders, then one more bandpass can be extracted from column 37, corresponding to edge (v_2, v_4). Since $35 = 2n^2 + (n - 1)$, graph G contains a Hamiltonian path and the induced Hamiltonian path from the row permutation is $\langle v_1, v_2, v_4, v_3 \rangle$.

For a fixed $B \geq 2$, we can make $B - 1$ more copies of those rows indexed $3i - 1$, for all i, and $B - 2$ copies of a new type of rows, each of them corresponds to an edge and has exactly one 1 in the column associated with the edge. This constructs an instance for the general Bandpass problem with a fixed bandpass number B. Similarly, it can be proven that graph G contains a Hamiltonian path if and only if this instance has a row permutation generating (at least) $2n^2 + (n-1)$ bandpasses. In summary, we have proved the following theorem.

Theorem 1 *The Bandpass problem with a fixed bandpass number $B \geq 2$ is NP-hard.*

3 Algorithmic results

Let n denote the number of columns in the Bandpass problem. The claim that *the Bandpass problem with $n \geq 3$ is NP-hard* is probably incorrect as made in [2]. Next, we extend the idea in the polynomial time exact algorithm for the special case $n \leq 2$ [2] to case $n = 3$ to obtain a solution which generates the maximum number of, or 1 less, bandpasses. Moreover, when $B = 2$, we guarantee to produce an optimal solution.

For $n = 1$, it is clear that putting the non-zero rows consecutively gives an optimal permutation, no matter what B is; For $n = 2$, firstly rows are classified into (0 0)-, (0 1)-, (1 0)-, and (1 1)-rows, then stacking them in the order of (0 0)-rows, then (0 1)-rows, then (1 1)-rows, lastly (1 0)-rows gives an optimal permutation, again no matter what B is. The optimality of the above two algorithms lies in the fact that all 1's in each column are placed consecutively. That is, for each column, the maximum number of bandpasses is achieved.

Such an idea can be extended for case $n = 3$, where we have 8 possible types of rows, (0 0 0)-, (0 0 1)-, (0 1 0)-, (0 1 1)-, (1 0 0)-, (1 0 1)-, (1 1 0)-, and (1 1 1)-rows. We stack the rows in the order of firstly (0 0 0)-rows, then sequentially (0 0 1)-rows, (0 1 1)-rows, (0 1 0)-rows, (1 1 0)-rows, (1 1 1)-rows, (1 0 1)-rows, and lastly (1 0 0)-rows. In this placement, the 1's in each of the first two columns appear consecutively, respectively, and the 1's in the third column are either consecutive as well or separated into two bands. Therefore, the number of bandpasses in this row stacking solution differs the optimum by at most 1.

There are many scenarios in which the achieved solution is optimal, for example, when there are no (0 1 0)- or (1 1 0)-rows, or when one of the two bands in the third column has size that is a multiple of B. In the special case of $B = 2$, if there is one column containing an odd number of 1's, then swapping this column into the third first, the row stacking solution is optimal.

In the other case, we tentatively assume that the achieved solution is non-optimal. Then, both $S_{101} + S_{111}$ and $S_{011} + S_{001}$ must be odd, where S_{101} denotes the number of (1 0 1)-rows (S_{111}, S_{011}, etc. are similarly defined). Due to $B = 2$, whenever there are two rows of the same type, we can remove them from consideration since putting them consecutively in the permutation achieves their maximum possible number of bandpasses. Therefore, we have the reduced instance in which one of S_{101} and S_{111} is 0, and the other is 1; similarly, one of S_{011} and S_{001} is 0, and the other is 1. Consider one case in which $S_{101} = 0$, $S_{111} = 1$, $S_{011} = 0$, and $S_{001} = 1$. If $S_{100} = 0$, then $S_{110} = 1$ and $S_{010} = 0$, and the row stacking obtained by swapping columns 1 and 3 is optimal. Otherwise, $S_{100} = 1$, and subsequently $S_{110} = 0$ and $S_{010} = 1$. This presents a scenario in which there is one (1 0 0)-row, one (0 1 0)-row, one (0 0 1)-row, and one (1 1 1)-row. It follows that the current solution is optimal, since there can't be a row permutation to generate three bandpasses. The other three cases can be similarly examined. The conclusion is, when $B = 2$, one of the six row stackings obtained from six column permutations of the matrix is optimal. These prove the following two theorems.

Theorem 2 *The Bandpass problem with $n = 3$ can be solved almost exactly in polynomial time, to obtain a row permutation generating either the maximum number of, or one less, bandpasses.*

Theorem 3 *The Bandpass problem with $n = 3$ and bandpass number $B = 2$ can be solved exactly in polynomial time.*

One instance of the Bandpass problem with $n = 3$ from [2] is listed in Figure 2(a). The stacking of the eight (for this instance, only six non-empty) types of rows is illustrated in Figure 2(b). We may also swap the first two columns to obtain another stacking (Figure 2(c)), which is optimal since in each column all 1's are placed consecutively. We conjecture that for any fixed bandpass number $B > 2$, by enumerating all possible 6 column permutations, one of the six row stackings of the corresponding eight types of rows would be an optimal solution.

	1	2	3		1	2	3		2	1	3
1	0	1	0	5	1	0	0	1	1	0	0
2	0	0	1	6	1	0	0	10	1	0	0
3	0	0	0	7	1	0	0	11	1	0	0
4	0	0	1	13	1	0	1	12	1	0	0
5	1	0	0	8	1	1	0	8	1	1	0
6	1	0	0	9	1	1	0	9	1	1	0
7	1	0	0	1	0	1	0	5	0	1	0
8	1	1	0	10	0	1	0	6	0	1	0
9	1	1	0	11	0	1	0	7	0	1	0
10	0	1	0	12	0	1	0	13	0	1	1
11	0	1	0	2	0	0	1	2	0	0	1
12	0	1	0	4	0	0	1	4	0	0	1
13	1	0	1	14	0	0	1	14	0	0	1
14	0	0	1	16	0	0	1	16	0	0	1
15	0	0	0	3	0	0	0	3	0	0	0
16	0	0	1	15	0	0	0	15	0	0	0

(a) (b) (c)

Figure 2: Solving an example instance of the Bandpass problem with $n = 3$: (a) the original instance, (b) the stacking of the eight types of rows, and (c) swapping columns 1 and 2 and the subsequent stacking of the eight types of rows.

Theorem 4 *There is an $O(B^2)$-approximation algorithm for the Bandpass problem with a fixed bandpass number $B \geq 2$.*

PROOF. We consider a further constrained Bandpass problem in which no two bandpasses in different columns can share any common rows. Essentially, a solution to this restricted version corresponds to a partition of all the m rows into chunks of exactly B rows, except the last chunk which might contain less than B rows. Given an optimal solution S^* to the original Bandpass instance, we can obtain a solution S_1^* to the restricted version by cutting out the bandpasses crossing rows 1 and 2, the bandpasses crossing rows $B+1$ and $B+2$, the bandpasses crossing rows $2B+1$ and $2B+2$, and so on. In fact, we can obtain in the similar way $B-1$ other solutions, $S_2^*, S_3^*, \ldots, S_B^*$. Clearly, every bandpass in S^* appears in exactly one of these B solutions. Consequently, $|S^*| = |S_1^*| + |S_2^*| + \ldots + |S_B^*|$, where $|S^*|$ denotes the number of bandpasses in S^*. It follows that the optimum value of the restricted version is at least $\frac{1}{B}$ of the optimum value of the original Bandpass problem.

Unfortunately we are not able to solve the above version of the partitioning problem exactly. While the general partitioning problem is NP-hard [5], our version can be described as a maximum weighted B-set packing problem [5]: An instance of the problem contains all sets of B distinct rows, each has a weight that is equal to the number of bandpasses in these B rows. The goal is to find a maximum weight collection of sets that are mutually disjoint. The maximum weighted B-set packing problem can be approximated within $B - 1 + \epsilon$ for any $\epsilon > 0$ [1] and within $\frac{2}{3}(B+1)$ [4]. Therefore, the Bandpass problem can be approximated within $O(B^2)$. □

Corollary 5 *There is a 2-approximation algorithm for the Bandpass problem with bandpass number $B = 2$.*

PROOF. From Theorem 1, the Bandpass problem with bandpass number $B = 2$ is NP-hard. The proof of the above Theorem 4 implies a 2-approximation algorithm, since the maximum weight bi-partitioning (equivalently, row-matching) problem can be solved in polynomial time. □

4 Conclusions

In this paper, the Bandpass problem with any fixed bandpass number $B \geq 2$ is shown NP-hard. A previous claim on the NP-hardness for the special case $n = 3$ is unlikely correct, because 1) a row stacking algorithm gives a solution generating at least one less the optimum bandpasses, and 2) it gives an optimal solution when the bandpass number B is 2. On approximation, an $O(B^2)$-algorithm for the general Bandpass problem is presented, which reduces to a 2-approximation algorithm for the Bandpass problem with bandpass number $B = 2$.

Acknowledgement

The research of GL was supported in part by NSERC.

References

[1] E. M. Arkin and R. Hassin. On local search for weighted packing problems. *Mathematics of Operations Research*, 23:640–648, 1998.

[2] D. A. Babayev, G. I. Bell, and U. G. Nuriyev. The bandpass problem: combinatorial optimization and library of problems. *Journal of Combinatorial Optimization*, 18:151–172, 2009.

[3] G. I. Bell and D. A. Babayev. Bandpass problem. In *Annual INFORMS meeting*, 2004. October 2004, Denver, CO, USA.

[4] B. Chandra and M. M. Halldórsson. Greedy local improvement and weighted set packing approximation. In *ACM-SIAM Proceedings of the Tenth Annual Symposium on Discrete Algorithms (SODA'99)*, pages 169–176, 1999.

[5] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman and Company, San Francisco, 1979.