Theoretical Computer Science 410 (2009) 963–967

Contents lists available at ScienceDirect

Theoretical Computer Science

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

Note

Approximation of the k-batch consolidation problem

Sung-Pil Hong [a], Myoung-Ju Park [a,*], Soo Y. Chang [b]

[a] Department of Industrial Engineering, Seoul National University, San 56-1 Shilim-Dong, Kwanahk Gu, Seoul, 151-742, Republic of Korea

[b] Department of Industrial and Management Engineering, Pohang University of Science and Technology, San 31 Hyoja-Dong, Pohang, Kyungbook, 790-784, Republic of Korea

ARTICLE INFO

Article history:
Received 23 June 2008
Accepted 5 November 2008
Communicated by D.-Z. Du

Keywords:
k-batch consolidation problem
Inapproximability
Approximation algorithm
k-set cover problem

ABSTRACT

We consider a problem of minimizing the number of batches of a fixed capacity processing the orders of various sizes on a finite set of items. This batch consolidation problem is motivated by the production system typical in raw material industries in which multiple items can be processed in the same batch if they share sufficiently close production parameters. If the number of items processed in a batch is restricted to being up to some fixed integer k, the problem is referred to as the k-batch consolidation problem. We will show that the k-batch consolidation problem admits an approximation whose factor is twice that of the k-set cover problem. In particular, this implies an upper bound on the approximation factor, $2H_k - 1$, where $H_k = 1 + \frac{1}{2} + \cdots + \frac{1}{k}$.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

Consider a production system where the orders $r(v) \in \mathbb{Q}_+$ on a finite set of items $v \in V$ are processed in batches. Each batch has a fixed capacity 1: the total order of items processed in a single batch cannot exceed 1. We are given a set, $E \subseteq V \times V$, of pairs of compatible items (u, v). Any set of items $S \subseteq V$ can be processed in the same batch if and only if they are compatible pairwise; in other words, S induces a clique on the compatibility graph $G = (V, E)$. Then naturally we can consider a problem of finding a minimum number of batches that can process the complete set of orders $\{r(v) : v \in V\}$. This problem will be referred to as the *batch consolidation problem* or *generalized batch consolidation problem* for emphasis. If there is an additional constraint that each batch cannot process more than k items for some constant $k \in \mathbb{Z}_+$, we call the problem the *k-batch consolidation problem*, which models the situation where proliferation of items in a single batch is prohibitive for a logistic reason.

Consider an integral version of the batch consolidation problem: given integer-valued orders $r(v) \in \mathbb{Z}_+$ and batch capacity $\lambda \in \mathbb{Z}_+$, the orders of items processed in the batches are required to be integer-valued. But, in [1], it has been observed that, when $k = 2$, given an optimal solution when allowed to process non-integral orders, one can construct the solution processing integral orders without increasing the number of utilized batches. Therefore it is an optimal solution of the integral version of the problem. It is not hard to show that such an observation extends to a general k. Thus, our definition of the batch consolidation problem using a unit batch capacity is general enough to cover the integral version.

The batch consolidation problem, first proposed by Lee et al. [7], was motivated by a production system typical for raw material industries such as steel, chemical and semiconductor ones. The process of a particular batch is characterized by a finite set of production parameters. Hence multiple items can be processed in the same batch if their parameters are sufficiently close. Naturally, the production efficiency depends on how well the batches are consolidated so that the number of utilized batches is minimized.

[*] Corresponding author. Tel.: +82 2 884 1164.
E-mail addresses: pmj0684@snu.ac.kr (M.-J. Park), syc@postech.ac.kr (S.Y. Chang).
URL: http://polytope.snu.ac.kr/sphong.html (S.-P. Hong).

0304-3975/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2008.11.007

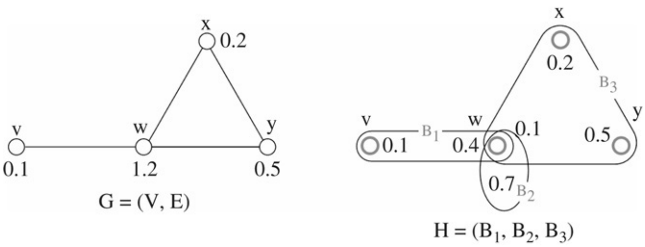

Fig. 1. The hypergraph determined by the batches B_1, B_2 and B_3.

It is not hard to see that the batch consolidation problem includes the clique partition problem [7,1], which implies that it does not admit an approximation within a factor of $|V|^\epsilon$ for some $\epsilon > 0$ [9]. But, as we will see, the k-batch consolidation problem is approximable within $2H_k - 1$ times the optimum, where $H_k = 1 + \frac{1}{2} + \cdots + \frac{1}{k}$. The idea is to decompose the orders of items so that a minimum cardinality set cover problem whose elements of the ground set correspond to the decomposed orders provides a well-consolidated set of batches. Note that this algorithm provides a 2-approximation when $k = 2$. Chang et al. [1] develop a $\frac{3}{2}$-approximation algorithm, for $k = 2$, based on a more elaborate scheme for the decomposition of orders of items. However, as also will be discussed later, once k becomes ≥ 3 such a scheme does not help in improving the approximation factor to strictly better than the one provided by the algorithm proposed in this paper.

The batch consolidation problem is related to *bin packing with conflicts*, or BPC [6,4]. Although a bin packing problem is fundamentally different from the batch consolidation problem as an item cannot be split over bins, BPC bears some similarity with the batch consolidation problem in that it specifies the pairs of items that cannot be packed in the same bin. [1] discusses some relations between the batch consolidation problem and BPC.

Another related model is *packing splittable items with cardinality constraints*, or PSIC [5]. PSIC is a generalization of bin packing: the items can be split over bins but a bin cannot contain more than k items. Notice, then, that PSIC [5] is the special case of the k-batch consolidation problem in which the compatibility graph G is complete. In this special case, the problem admits a polynomial time approximation scheme while, in general, the problem is max-SNP-hard and not approximable within 1.0021 times the optimum as discussed later.

This paper is organized as follows. Section 2 discusses a simple but useful property of an optimal solution helpful in the analysis of the approximation algorithm. In Section 3, we establish an inapproximability of the k-batch consolidation problem. Section 4 is devoted to the discussion of an approximation algorithm.

2. Preliminaries

Given a solution of the problem, consider the hypergraph $\mathcal{H} = (V, \mathcal{B})$ whose vertices and edges, respectively, correspond to the items V and the collection of batches $B \in \mathcal{B}$ processing items with their nonzero orders. (See Fig. 1.)

Proposition 2.1. *Any solution of the k-batch consolidation problem can be modified efficiently without increasing the number of utilized batches so that its hypergraph $\mathcal{H}$ is acyclic: there is no sequence $(v_1, B_1, v_2, B_2, \ldots, v_l\, B_l, v_1)$ with $l \geq 2$ such that B_i are all distinct, v_i are all distinct, and $v_i, v_{i+1} \in B_i$ for $i = 1, 2, \ldots, l-1$, and $v_l, v_1 \in B_l$. (We refer to such sequence as a* circuit *of a hypergraph.)*

Proof. Suppose $\mathcal{H}$ has a circuit $C := (v_1, B_1, v_2, B_2, \ldots, v_l\, B_l, v_1)$. For each v_i, $i = 1, 2, \ldots, l$, from C, write, as ρ_i^- and ρ_i^+, respectively, the orders of v_i processed by B_{i-1} and B_i batches where B_0 denotes B_l. Reverse the direction and/or redesignate the initial vertex of circuit so that $\min_{v_i \in C}\{\rho_i^-, \rho_i^+\} = \rho_1^+$. Then the modified solution $\rho_i^+ - \rho_1^+, \rho_i^- + \rho_1^+, \forall v_i \in C$ is feasible with the same number of batches. And the circuit is disconnected since v_1 is deleted from B_1. Repeating this procedure, we can convert any solution to having an acyclic hypergraph. □

Proposition 2.2. *Any problem (G, r) can be reduced in polynomial time into one, (G, s), with $s(v) < \deg(v) + 1, \forall v \in V$.*

Proof. From Proposition 2.1, there is an optimal solution which has at most $\deg(v)$ batches processing the order of $v \in V$ with other items. In other words, when $r(v) \geq \deg(v) + 1$, we can first construct $\lfloor r(v) - \deg(v) \rfloor$ batches processing exactly 1 out of the order $r(v)$ without compromising the optimality. Then the orders are reduced to $s(v) = r(v) - \lfloor r(v) - \deg(v) \rfloor < \deg(v) + 1$. □

3. Inapproximability

We can derive an easy inapproximability of the k-batch consolidation problem from the inapproximability of the 2-batch consolidation problem by Chang et al. [1]. In the reduction from the vertex cover problem with bounded degree to an instance of the 2-batch consolidation problem, they construct the compatibility graph G as bipartite. A k-batch consolidation problem on a bipartite G is simply a 2-batch consolidation problem and hence the reduction is also valid for the k-batch consolidation problem.

Theorem 3.1. *The k-batch consolidation problem cannot be approximated within* 1.0021 *times the optimum for all $k \geq 2$ unless* P = NP.

Proof. Chang et al. [1] prove that if a vertex cover problem with δ-bounded degree cannot be approximated within ρ, then the k-batch consolidation problem cannot be approximated within $1+\frac{\rho-1}{2\delta+1}$. And Chlebik and Chlebikova [3] show that the vertex cover problem with 4-bounded degree cannot be approximated within $\frac{53}{52}$. Therefore the k-batch consolidation problem cannot be approximated within 1.0021. □

4. Approximation

4.1. Set-cover-based algorithm

For a problem (G, r) defined by $G = (V, E)$ and $r \in \mathbb{Q}_+^V$, the approximation algorithm is conveniently described by defining an auxiliary problem $(U, \mathcal{S})$ where U and $\mathcal{S}$ are constructed as follows. For each $v \in V$, compute $n_v := \lceil kr(v) \rceil$ and accordingly construct a decomposition of $r(v)$: a set of n_v elements, $D_v = \{v_0, v_1, v_2, \ldots, v_{n_v-1}\}$ and their orders $r'(v_0) = r(v) - \frac{n_v-1}{k}, r'(v_1) = r'(v_2) = \cdots = r'(v_{n_v-1}) = \frac{1}{k}$. Let $U = \bigcup_{v \in V} D_v$ and $\mathcal{S} = \{S \subseteq U | \sum_{u \in S} r'(u) \leq 1, |S| \leq k, \ S \text{ is a clique}\}$. The following $(U, \mathcal{S})$ is an auxiliary problem of instance $G = (V, E)$ in Fig. 1 for $k = 2$.

$$
\begin{aligned}
U = {} & \{v_0, w_0, w_1, w_2, x_0, y_0\} \\
\mathcal{S} = {} & \{\{v_0\}, \{w_0\}, \{w_1\}, \{w_2\}, \{x_0\}, \{y_0\} \\
& \{v_0, w_0\}, \{v_0, w_1\}, \{v_0, w_2\} \\
& \{w_0, w_1\}, \{w_0, w_2\}, \{w_0, x_0\}, \{w_0, y_0\} \\
& \{w_1, w_2\}, \{w_1, x_0\}, \{w_1, y_0\} \\
& \{w_2, x_0\}, \{w_2, y_0\}, \{x_0, y_0\}\}.
\end{aligned}
$$

In the above, $|U| = \sum_{v \in V} \lceil kr(v) \rceil \leq \sum_{v \in V} \lceil |V| r(v) \rceil < \sum_{v \in V} (1 + |V| r(v)) \leq |V| + |V| \sum_{v \in V} r(v)$. But, due to Proposition 2.2, we have $r(v) < 1 + \deg(v), \forall v \in V$, which implies $|U| < |V| + |V| \sum_{v \in V} (1 + \deg(v)) = |V| + |V|^2 + 2|V||E|$. As $|\mathcal{S}| \leq \sum_{j=1}^{k} |U|^j$, the construction can be performed in polynomial time for a fixed k.

Consider the following algorithm of the k-batch consolidation problem.

Algorithm 4.1.

Step 1 Construct the auxiliary problem $(U, \mathcal{S})$ of (G, r).
Step 2 Compute a minimum set cover $\mathcal{C} \subseteq \mathcal{S}$ of U.
Step 3 For each subset C of $\mathcal{C}$, construct a batch processing the assigned orders, $r'(u), \forall u \in C$. Return the batches as a solution.

From the construction of $(U, \mathcal{S})$, the batches from Step 3 can cover the complete set of orders. Also notice that we can adjust the processing orders without increasing the number of batches so that each order $r'(u)$ is exactly covered. As the orders r' of the auxiliary problem are a decomposition of the original orders r, the batches from Step 3 are clearly a feasible solution of the original problem.

Let us define some more notation. Denote by $OPT(G, r)$ and $z(G, r)$, respectively, the numbers of batches of an optimal solution and a solution returned by Algorithm 4.1. And $c(X)$ is the cardinality of a minimum set cover from $\mathcal{S}$ of $X \subset U$. Then, for the analysis of Algorithm 4.1, the following lemmas are useful.

Lemma 4.2. *For any partition $(X; Y)$ of U, $z(G, r) = c(U) \leq c(X) + c(Y)$.*

Proof. Let $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively, be the minimum set covers of X and Y. Then, $\mathcal{C}_1 \cup \mathcal{C}_2$ is a set cover of U. Therefore,

$$z(G, r) = c(U) \leq |\mathcal{C}_1| + |\mathcal{C}_2| = c(X) + c(Y). \quad \square$$

Lemma 4.3. *If $r \leq s$, $OPT(G, r) \leq OPT(G, s)$ and $z(G, r) \leq z(G, s)$.*

Proof. The first half of the statement is trivial.

For the second half, define r such that $r(w) = s(w) - \delta$, with $0 < \delta \leq \frac{1}{k}$ for any fixed $w \in V$ and $r(v) = s(v), \forall v \in V \backslash \{w\}$. Let $(U_r, \mathcal{S}_r)$ and $(U_s, \mathcal{S}_s)$, respectively, be the auxiliary problems for (G, r) and (G, s). Also let $D_w = \{w_0, w_1, \ldots, w_l\}$ be a set for w in the decomposition of $s(w)$ from (G, s).

If $s'(w_0) > \delta$ in the decomposition of $s(w)$, then $U_r = U_s$ and hence $\mathcal{S}_r \supseteq \mathcal{S}_s$. Therefore a set cover of $(U_s, \mathcal{S}_s)$ is also a set cover of $(U_r, \mathcal{S}_r)$ and we have $z(G, r) \leq z(G, s)$.

If, on the other hand, $s'(w_0) \leq \delta$, then we get $D_w = \{w_0, w_1, \ldots, w_{l-1}\}$ for w in the decomposition of $r(w)$ from (G, r). Let $\mathcal{C}_s$ be any set cover of U_s. Delete w_0 from the subset of $\mathcal{C}_s$. And replace w_l with w_0. Then the resulting collection is a set cover of U_r whose cardinality is no greater than $|\mathcal{C}_s|$. (See Fig. 2.) Therefore we have $z(G, r) \leq z(G, s)$. Repeating this procedure we can prove $z(G, r) \leq z(G, s)$ for any case of $r \leq s$. □

Theorem 4.4. *$z(G, r) \leq 2OPT(G, r)$.*

Proof. By induction on $OPT(G, r)$. Suppose $OPT(G, r) = 1$. Then G is a complete graph with $|V| \leq k$ and $\sum_{v \in V} r(v) \leq 1$. This implies that both $S_1 = \{u \in U | r'(u) < \frac{1}{k}\}$ and $S_2 = \{u \in U | r'(u) = \frac{1}{k}\}$ are elements of $\mathcal{S}$. But, $\{S_1, S_2\}$ is a set cover of U and we have $z(G, r) \leq 2 = 2OPT(G, r)$.

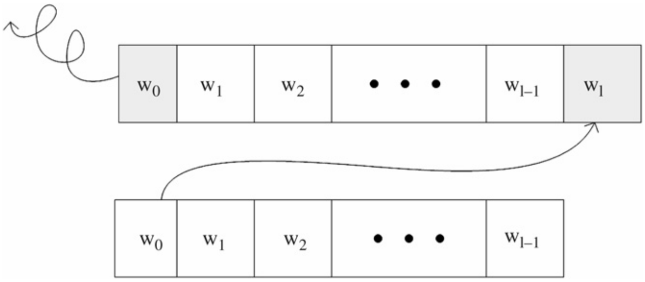

Fig. 2. Modifying a set cover of U_s to that of U_r.

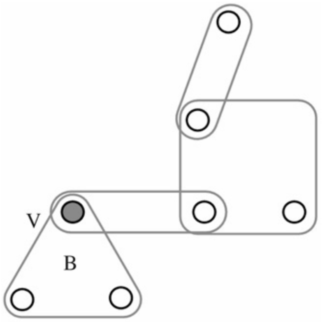

Fig. 3. The case $r(v)$ is split over two batches.

Assume that the theorem holds for any problem whose optimal batch number is less than n and consider any problem (G, r) with $OPT(G, r) = n$. Let $S(G, r)$ be a corresponding optimal solution. From Proposition 2.1, the hypergraph H determined by $S(G, r)$ can be assumed to be acyclic. Therefore there is a batch B which has at most one item $v \in V$ whose order $r(v)$ is split over more than one batch.

Suppose B has no such split item. Then define $X := \bigcup_{w \in B} D_w$ and $Y := U \setminus X$. Let s and t be the vectors obtained by restricting r to X and Y, respectively. Then we get $OPT(G, s) = 1$ and $OPT(G, t) = n - 1$. From Lemma 4.2 and the induction hypothesis, we have

$$z(G, r) \leq c(X) + c(Y) \leq 2 + 2(OPT(G, r) - 1) = 2OPT(G, r).$$

Now suppose B has such a split item $v \in B$ (see Fig. 3). And B processes the order $r_B(v) := \frac{l}{k} + \delta$ from $r(v)$ for some $l \in \mathbb{Z}_+$ and $0 \leq \delta < \frac{1}{k}$. Let $D_v = \{v_0, v_1, v_2, \ldots, v_{n_v}\}$ be a set for v in the decomposition of $r(v)$. Note that $n_v \geq l + 1$ if $\delta > r'(v_0)$. Define

$$A := \begin{cases} \{v_1, \ldots, v_l\}, & \text{if } \delta = 0, \\ \{v_0, v_1, \ldots, v_l\}, & \text{if } r'(v_0) \geq \delta > 0, \\ \{v_1, \ldots, v_{l+1}\}, & \text{if } \delta > r'(v_0), \end{cases}$$

and $X := \bigcup_{w \in (B \setminus v)} D_w \cup A$ and $Y := U \setminus X$. Then notice that $0 \leq \sum_{u \in A} r'(u) - r_B(v) < \frac{1}{k}$ and therefore we have $\sum_{u \in X} r'(u) < 1 + \frac{1}{k}$. This implies that there are at most k elements u of X such that $r'(u) = \frac{1}{k}$. Also $|B| \leq k$ implies that no more than k elements u of X have $r'(u) < \frac{1}{k}$. Thus, if we set $S_1 := \{v \in X | r'(v) = \frac{1}{k}\}$, and $S_2 := \{v \in X | r'(v) < \frac{1}{k}\}$, then $S_1, S_2 \in \mathcal{S}$ and $\{S_1, S_2\}$ is a set cover of X and we have $c(X) \leq 2$.

Let s and t be the vectors obtained by subtracting from r the orders corresponding to the elements of X and B, respectively. Then, $s \leq t$ and hence from Lemma 4.3, $OPT(G, s) \leq OPT(G, t) = OPT(G, r) - 1$ and $z(G, s) \leq z(G, t)$. But, then from the induction hypothesis, $c(Y) = z(G, s) \leq z(G, t) \leq 2(OPT(G, r) - 1)$. Finally, from Lemma 4.2, we have

$$z(G, r) \leq c(X) + c(Y) \leq 2 + 2(OPT(G, r) - 1) = 2OPT(G, r). \quad \square$$

A tight example of Theorem 4.4. Consider a complete graph $Q = (W, A)$ with $|W| = k$ and an order vector $s \in \mathbb{Q}^k_+$ where $s(v_1) = \frac{1}{k} - \epsilon(k - 1)$, and $s(v) = \frac{1}{k} + \epsilon, \forall v \in W \setminus \{v_1\}$. Then, $OPT(Q, s) = 1$ and $z(Q, s) = 2$.

Construct a graph G by connecting l such Q's via a path consisting of v_1 of each Q. The order $r \in \mathbb{Q}^{kl}_+$ of G is simply the direct sum of l identical order vectors $s \in \mathbb{Q}^k_+$. Then, as easily checked, $OPT(G, r) = l$ and $z(G, r) = 2l$. Hence the analysis of Theorem 4.4 is tight.

Step 2 of Algorithm 4.1, however, cannot be performed in polynomial time as the k-set cover problem is NP-hard. We can, however, rely on an approximation algorithm of the k-set cover problem. For instance, the approximation algorithm of [2] guarantees the approximation factor of $(H_k - \frac{1}{2})$, where $H_k = 1 + \frac{1}{2} + \cdots + \frac{1}{k}$. Thus, employing the approximate solution instead of an exact one, Algorithm 4.1 is an $(2H_k - 1)$-approximation. For $k \geq 4$, we can use the $(H_k - \frac{196}{390})$-approximation algorithm of [8] instead to get a slightly improved $(2H_k - \frac{196}{195})$-approximation for the k-batch consolidation problem.

4.2. *An alternative decomposition scheme*

When $k = 2$, Algorithm 4.1 employing an $(H_k - \frac{1}{2})$-approximation of the k-set cover problem provides a 2-approximation as the corresponding 2-set cover problem is nothing but than the polynomially solvable minimum edge-cover problem. But, the specialized 2-batch problem algorithm of [1] guarantees the approximation factor, $\frac{3}{2}$. The algorithm is based on the same idea of solving the edge-cover problem on the auxiliary problem obtained by decomposition of the orders of vertices. But, it uses a slightly different decomposition: each vertex v of order $r(v)$ is decomposed into $2 \times \lfloor r(v) \rfloor$ vertices all assigned the order $\frac{1}{2}$, and one vertex of the order $r(v) - \lfloor r(v) \rfloor$ in the auxiliary problem. The remaining steps are exactly the same as Algorithm 4.1 for $k = 2$. Thus there is one (and at most one) auxiliary vertex per original one, whose order can be greater than $\frac{1}{2}$. As shown in [1], when the number of items processed in a single batch is restricted to a number as small as $k = 2$, such vertices are crucial in attaining the approximation guarantee of $\frac{3}{2}$.

Interestingly enough, when $k \geq 3$, however, such a decomposition scheme does not help in improving the approximation guarantee to strictly better than 2. To see this, consider the complete graph $G = (V, E)$ with $|V| = 2l - 1$ and $r(v) = \frac{1}{2} + \varepsilon$, $\forall v \in V$. Then, for each $v \in V$, we get $r'(v_0) = \frac{1}{2} + \varepsilon$ and therefore v_0 participates only in a singleton set in the auxiliary problem. Thus $z(G, r) = 2l - 1$ while the optimal value of the k-batch problem is $OPT(G, r) = \lceil (2l - 1)(\frac{1}{2} + \varepsilon) \rceil = l$ for all $k \geq 3$. Thus the approximation factor is $2 - \frac{1}{l}$.

5. Further research

There is currently a significant gap between the upper bound $(2H_k - 1)$ and the lower bound 1.0021 on the approximability of the k-batch consolidation problem. It is an interesting open problem whether the lower bound can be tightened to $\log k$, asymptotically the same as the upper bound, or vice versa.

Acknowledgement

The second author's research was partially supported by the Second Stage of Brain Korea 21 Project in 2008.

References

[1] J. Chang, S.Y. Chang, S.-P. Hong, Y.-H. Min, M.-J. Park, Approximation of a batch consolidation problem, 2008 (submitted for publication).
[2] R. chii Duh, M. Fürer, Approximation of k-set cover by semi-local optimization, in: Proceedings of the 29th ACM Symposium on Theory of Computing, ACM, 1997, pp. 256–264.
[3] M. Chlebik, J. Chlebikova, Complexity of approximating bounded variants of optimization problems, Theoretical Computer Science 354 (3) (2006) 320–338.
[4] L. Epstein, A. Levin, On bin packing with conflicts, in: Proceedings of the 4th Workshop on Approximation and Online Algorithms, in: Lecture Notes in Computer Science, vol. 4368, Springer, 2006, pp. 160–173.
[5] L. Epstein, R. van Stee, Approximation schemes for packing splittable items with cardinality constraints, in: Proceedings of the 5th Workshop on Approximation and Online Algorithms, in: Lecture Notes in Computer Science, vol. 4927, Springer, 2007, pp. 232–245.
[6] K. Jansen, An approximation scheme for bin packing with conflicts, in: Proceedings of the 6th Scandinavian Workshop on Algorithm Theory, in: Lecture Notes in Computer Science, vol. 1432, Springer, 1998, pp. 35–46.
[7] K. Lee, S.Y. Chang, Y. Hong, Continuous slab caster scheduling and interval graphs, Production Planning and Control 15 (5) (2004) 495–501.
[8] A. Levin, Approximating the unweighted k-set cover problem: Greedy meets local search, in: Proceedings of the 4th Workshop on Approximation and Online Algorithms, in: Lecture Notes in Computer Science, vol. 4368, Springer, 2006, pp. 290–301.
[9] C. Lund, M. Yannakakis, On the hardness of approximating minimization problems, Journal of ACM 41 (5) (1994) 960–981.