Approximation of a Batch Consolidation Problem

Junho Chang and Soo Y. Chang
Department of Industrial and Management Engineering, Pohang University of Science and Technology, San 31 Hyoja-Dong, Pohang, Kyungbook 790-784, Republic of Korea

Sung-Pil Hong, Yun-Hong Min, and Myoung-Ju Park
Department of Industrial Engineering, Seoul National University, San 56-1 Shilim-Dong, Kwanahk Gu, Seoul, 151-742, Republic of Korea

In batch production systems, multiple items can be processed in the same batch if they share sufficiently similar production parameters. We consider the batch consolidation problem of minimizing the number of batches of a finite set of items. This article focuses on the case in which only one or two items can be processed in a single batch. The problem is NP-hard and cannot be approximated within 1.0021 of the optimum under the premise, P $\neq$ NP. However, the problem admits a $\frac{3}{2}$-approximation. The idea is to decompose the demands of items so that a maximum matching in the graph on the vertices of the decomposed demands provides a well-consolidated batch set. **NETWORKS, Vol. 58(1), 12–19 2011**

Keywords: batch production; minimization of batch number inapproximability; approximation algorithm

1. INTRODUCTION

Consider a production system of a finite set of items, $v \in V$ whose demands $r(v) \in \mathbb{Q}_+$ are processed in batches. Each demand $r(v)$ can be split into multiple batches, but none of the batches can receive a total demand exceeding its capacity $\lambda \in \mathbb{Q}_+$. The production systems includes a set of pairs of compatible items $(u, v) \in E \subseteq V \times V$. A set of items can be processed in the same batch if and only if they are all pairwise compatible. In this article, we consider a variant of this problem in which the number of compatible items that can be processed in the same batch does not exceed two. Thus, each batch corresponds to a pair $(u, v) \in E$ or a single item $v \in V$. Our problem is to find the minimum number of batches to process the demands $\{r(v) : v \in V\}$.

This batch consolidation problem (BCP) is motivated by production in raw material industries in which demands of various sizes are typically processed in batches. The processing of a particular batch is characterized by a finite set of production parameters; hence multiple demands, even though they are not on the same item, can be processed in the same batch if their parameters are sufficiently similar. Naturally, production efficiency depends on how well the batches are consolidated so that the number of utilized batches is minimized. For instance, [3, 13] addressed to the optimization of such a production arising in the steel industry.

The BCP was first proposed by Lee et al. [8] in the context of slab caster scheduling in a steel mill. This original formulation had no restriction on the number of items processed in a batch; therefore, it will be called the generalized batch consolidation problem (GBCP). The GBCP is NP-hard and can be solved polynomial time when the items that can be processed in the same batches determine an interval graph [8]. For a clear distinction between the problems if the number is restricted to be $\leq k$, the problem will be called a k-batch consolidation problem (k-BCP).

GBCP includes the clique partition problem as a special case, because, when λ is large enough to cover all the demands of any clique, each clique partition of the underlying graph defines a feasible solution [8]. Conversely, any feasible set of batches can be modified so that no two batches share a common item, without increasing the number of batches. Therefore, each solution defines a clique partition of the underlying graph. The clique partition problem cannot be approximated within $|V|^\epsilon$ for some $\epsilon > 0$ [9] (hence, it is a "Class-IV" problem [1]). Therefore, GBCP seems to be a very difficult problem even in the sense of approximability.

The GBCP is fundamentally different from the bin packing problem in which items are not divisible into bins. Among the extensions of the bin packing problem, the bin packing with conflicts [6] is similar to GBCP in the sense that

Received January 2008; accepted June 2010
Correspondence to: S.-P. Hong; e-mail: sphong@snu.ac.kr
Contract grant sponsors: Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2009-0077035). The first, fourth, and fifth authors were also partially supported by the second stage of Brain Korea 21 project in 2009.
DOI 10.1002/net.20409
Published online 1 October 2010 in Wiley Online Library (wileyonlinelibrary.com).

it specifies the pairs of items that cannot be packed in the same bin. When the bin size is sufficiently larger than the item sizes, this problem reduces to the coloring problem on the graph whose vertices and edges, respectively, represent the items and their conflicting pairs. Thus the bin packing with conflicts as well as GBCP can be considered to be generalizations of the clique partition problem. However, the bin packing problems seems harder than GBCP because the bin packing itself is already an NP-hard problem, and therefore, it is NP-hard on any class of graphs containing graphs with an empty edge set, such as the comparability graphs, which are the complements of the interval graphs. However, GBCP is polynomially solvable on the interval graphs [8]. However, if we restrict the maximum number of items packed in a bin analogously to k-BCP, the observation does not seem to extend straightforwardly: if the maximum number is two, bin packing with conflicts reduces to a maximum cardinality matching problem, whereas 2-BCP is Max-SNP-hard as will be proved. The approximability of the bin packing with conflicts is known for some special graphs: a constant factor approximation for perfect graphs [6] and an asymptotic approximation scheme for a d-inductive graphs with a constant d [7]. The best known approximation algorithms for the problem on the perfect graphs and some classes of graphs are given in Ref. [4].

When k is fixed, the possible sets of items per batch can be enumerated in polynomial time. Thus, k-BCP can be considered as the capacitated variant of the set cover problem in which multiple copies of a set can be used and in which each copy provides an additional capacity, λ that can be shared by the items in the set. This suggests that, for a fixed k, k-BCP can be solved using a set-covering problem algorithm combined with a scheme of assigning λ to the demands of items of a set.

Recently, an approximation algorithm has been developed [5] that solves the minimum cardinality k-set cover problem on an auxiliary problem obtained by decomposing the demands $r(v)$, $v \in V$, a priori, in a manner similar with but more straightforward than the one adopted in this article. Such a solution has an approximation factor within twice that of the corresponding minimum cardinality k-set cover solution. However, without such a prior decomposition, a greedy algorithm for the ordinary set covering can be extended to this case to guarantee an equivalent approximation factor, $2H_k$ where $H_k = 1 + 1/2 + \cdots + 1/k$.

This article explores in/approximability of 2-BCP. Theoretically, compared to the GBCP, which is at least as hard as the Class VI problems, the BCP seems to have convenient properties that admit various approximation approaches. Perhaps more importantly, establishing a tight lower or upper bound on the approximability of the BCP may be a step to solving more general problems such as k-BCPs, with a small k, or GBCP.

The BCP includes the maximum cardinality matching problem as a special case. If λ is large enough to cover the demands of any pair of items, then any set M of mixed batches may be assumed to define a matching, and the total number of batches is given by $|V| - |M|$. Hence, the problem is reduced to a maximum cardinality matching problem, a well-known polynomially solvable problem [12]. However, generally, the BCP is NP-hard and does not admit a polynomial time approximation scheme as will be proved by a reduction from the vertex cover problem with bounded degree.

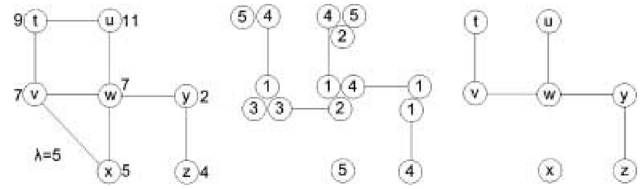

FIG. 1. An instance (G, r), a feasible solution (S, r), and its solution graph.

Nevertheless, the matching problem offers an essential ingredient of the $\frac{3}{2}$-approximation algorithm for 2-BCP proposed in this article. The idea is to decompose the demands of items to obtain a graph whose vertices correspond to the decomposed demands such that a maximum matching provides a well-consolidated set of batches. In doing so, the size of the auxiliary problem is polynomially bounded in the problem size.

This article is organized as follows. Section 2 introduces the notation and a set of propositions that are useful in the discussion of the inapproximability and the approximation algorithm. Section 3 is devoted to the inapproximability of the batch consolidation problem: we reduce the vertex cover problem with bounded degree into the batch consolidation problem to show that the latter cannot be approximated within 1.0021 of the optimum. Finally, Section 4 describes the $\frac{3}{2}$-approximation algorithm.

2. NOTATION AND PRELIMINARIES

When the batch capacity λ is given, each problem instance defines an undirected graph $G = (V, E)$ with the demand function $r : V \to \mathbb{Q}_+$. Thus, each problem will be denoted by a pair (G, r) (Fig. 1). We may assume that G is connected.

We use $V(H)$ and $E(H)$, respectively, to denote the vertex and edge set of the graph H. We use the standard notation $H[U]$ to denote the subgraph of H induced by its vertex subset $U \subseteq V(H)$. However, for the problem-defining graph G, we reserve the shorthand notation V and E, respectively, instead of $V(G)$ and $E(G)$. By χ^S we mean the characteristic vector of S whose dimension will be understood from the context. If S is singleton, $S = \{x\}$, we will simply write χ^x instead of $\chi^{\{x\}}$.

A feasible solution $S(G, r)$(Fig. 1) of the batch consolidation problem (G, r) is specified by the set of batches processing nonzero demands, each of which is, in turn, specified by the two items processed in the batch and by the demands of the items processed in it. If the demand of one item in a batch is zero, the batch is called pure, otherwise it is called and mixed. The batch will be called saturated if the sum of demands processed is equal to λ, and nonsaturated, otherwise. Given a feasible solution $S(G, r)$, its solution graph is defined on V where uv is an edge of the solution graph if and only if there exists a mixed batch processing the (nonzero) demands of both u and v.

The following observations will be useful in the subsequent discussion.

Proposition 2.1. *Given any feasible solution, we can find a solution in polynomial time satisfying the following:*

1. *No vertex has more than one nonsaturated pure batch.*
2. *No edge has more than one mixed batch.*

Proof. The first statement is trivial. Suppose $uv \in E$ has two mixed batches and let $\rho(v)$ be the total demand on v processed in the two batches. If $\rho(u)+\rho(v) \leq \lambda$, then clearly the two batches can be merged into one. If $\lambda < \rho(u)+\rho(v) \leq 2\lambda$ and $\rho(u) \leq \rho(v)$, then the same demands can be covered by a saturated mixed batch processing the demands $\rho(u)$ on u and $\lambda - \rho(u)$ on v, and a pure batch processing demand $\rho(u) + \rho(v) - \lambda$ on v. If $\rho(u) > \rho(v)$, we can use a similar argument to show the proposition. ∎

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

Proof. From Proposition 2.1, an optimal solution exists in which at most $\deg(v)$ mixed batches are used for $v \in V$. Therefore, when $r(v) \geq (\deg(v)+1)\lambda$, we can first construct $\lfloor \frac{r(v)-\lambda \deg(v)}{\lambda} \rfloor$ saturated pure batches out of the demand $r(v)$ without compromising the optimality. Then, the demands are reduced to $r'(v) = r(v) - \lambda \lfloor \frac{r(v)-\lambda \deg(v)}{\lambda} \rfloor < (\deg(v)+1)\lambda$. ∎

Proposition 2.3. *Any solution can be modified in polynomial time without increasing the number of batches such that its solution graph is a forest.*

Proof. Let a solution graph have a circuit C. Give an orientation on C counterclockwise. For each $v \in V(C)$, let $\rho_+(v)$ and $\rho_-(v)$ be, respectively, the demands processed by the batches corresponding to outgoing and incoming arcs to v. Then, for every ϵ, the modified solution $\rho_+(v)+\epsilon$, $\rho_-(v) - \epsilon \forall v \in V(C)$ is also feasible if $\rho_+(v) + \epsilon \geq 0$, $\rho_-(v) - \epsilon \geq 0 \ \forall v \in V(C)$. Thus, if we set

$$\epsilon = \begin{cases} \min_v\{\rho_+(v), \rho_-(v) : v \in V(C)\}, & \text{if the minimum is attained in an incoming arc,} \\ -\min_v\{\rho_+(v), \rho_-(v) : v \in V(C)\}, & \text{otherwise,} \end{cases}$$

at least one arc on C will be removed from the solution graph. Repeating this procedure, we can convert the solution graph into an acyclic graph. ∎

The following proposition shows that the integral version of the problem is the same as the original problem.

Proposition 2.4. *Suppose that the demands and λ are all integers. Then any solution can be modified in polynomial time without increasing its number of batches so that the processed demands are all integers.*

Proof. Apply to a given solution, the procedures from Proposition 2.3 and 2.1 in order. The acyclic property from Proposition 2.3 is maintained during the procedure from Proposition 2.1. Then, an isolated vertex of an item, if any, should receive an integral demand due to the integrality assumption.

Consider the subgraph F of the solution graph induced by the fractional batches, namely, the batches that received at least one noninteger demand of an item. Then, F should be a forest with no isolated vertex.

Consider any vertex u of degree one. If the demand of u is split fractionally into two batches, one of them is a fractional pure batch and hence by transferring the fractional part of the demand from the mixed batch, we can reduce the number of fractional batches at least by one without increasing the number of batches.

If the demand u is not processed fractionally in any batch, then there exists a mixed batch uv processing a fractional demand $\rho(v)$ of v and the demand $r(v)$ should be split fractionally into other batches. Then by transferring from such batches, we can increase the demand of v processed by the mixed batch uv to an integer. The number of fractional batches is decreased at least by one without increasing the number of batches.

Repeating above procedure, we can modify a solution without increasing its number of batches so that the processed demands are all integers. ∎

3. INAPPROXIMABILITY

The following degree-bounded version of the vertex cover problem is useful in establishing inapproximabilities of combinatorial optimization problems.

Problem 3.1. *Vertex Cover with Bounded Degree (VCB)*
INPUT: *An undirected graph $G = (V, E)$ and $\delta \in \mathbb{Z}_+$ such that* $\deg(v) \leq \delta \forall v \in V$.
OPTIMIZATION: *Find a vertex cover of G with minimum cardinality.*

Theorem 3.2. *The batch consolidation problem cannot be approximated within* 1.0021 *times the optimum unless* P = NP.

Proof. Consider the following reduction from an instance I of VCB given by $G = (V, E)$ and δ to an instance $f(I)$ given by $(G' = (V', E'), r)$ with the capacity λ of the batch consolidation problem. $\lambda = \delta + 1$. The graph G' is obtained from the subdivision of each edge of G into 4 edges: for each $e = uv \in E$, define a vertex set $V'_e = \{u_e, m_e, v_e\}$. Also, for each $e = uv \in E$, construct $E'_e = \{uu_e, u_e m_e, m_e v_e, v_e v\}$. Define $V' = V \cup (\bigcup_{e \in E} V'_e)$ and $E' = \bigcup_{e \in E} E'_e$. Finally, for each $e = uv \in E$, let

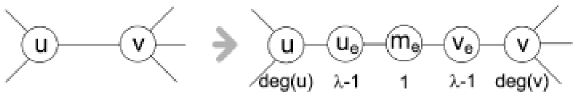

FIG. 2. Subdivision of each edge and the demands of vertices.

$r(u) = \deg(u)$, $r(v) = \deg(v)$, $r(u_e) = r(v_e) = \lambda - 1$, and $r(m_e) = 1$(Fig. 2). The reduction can be performed in a linear time of the number of edges.

We now show that the optimal values of an instance of VCB and its reduction are polynomially related: the former has a vertex cover of size $\leq k$ if and only if the latter has a feasible solution whose number of batches is $\leq 2|E| + k$.

Suppose I has a vertex cover C. For each $uv \in E$, construct a feasible solution of $f(I)$ as follows.

CASE 1. $u \in C$ and $v \notin C$. Then cover $r(u)$ with a pure batch, $r(u_e)$ and $r(m_e)$ with a single mixed batch, and $r(v_e)$ and a unit demand from $r(v)$ with a mixed batch.

CASE 2. $u, v \in C$. Cover $r(u)$, $r(v)$, and $r(v_e)$ with three pure batches, and $r(u_e)$ and $r(m_e)$ with one mixed batch.

Notice that for each $v \notin C$, its demand, $r(v) = \deg(v)$, is split into exactly $\deg(v)$ mixed batches. Therefore, the resulting solution is feasible and the total number of batches used is $|C| + 2|E|$. This implies

$$\mathrm{OPT}(f(I)) \leq \mathrm{OPT}(I) + 2|E|. \tag{1}$$

Conversely, suppose that $S(G', r)$ is any feasible solution of $f(I)$. We may assume that the processed demands in $S(G', r)$ are all integers(Proposition 2.4). Let $E'_e = \{uu_e, u_em_e, m_ev_e, v_ev\}$ be any subdivision of G'. Consider the batch processing the unit demand of m_e. If the batch is mixed and is also processing any adjacent vertex, say, u_e, then we can saturate it, if necessary, by transferring the demand of u_e from other batch(es) processing u_e. If it is a pure batch, we can also saturate it by transferring similarly after choosing any adjacent vertex. Clearly this process does not increase the number of batches.

So far the demands of u_e and m_e have been consolidated in a single batch. Also from Proposition 2.1, we may assume the demand of v_e is split into at most two batches, one pure and one mixed. Thus if it is split into two, we can merge the split demands of v_e into the pure batch (because $r(v_e) = \lambda - 1$). Thus, we can always modify $S(G', r)$ in polynomial time without increasing the number of batches so that the demands of $r(u_e) = \lambda - 1$, $r(m_e) = 1$, and $r(v_e) = \lambda - 1$ are covered by two batches for any $e \in E$. We will call such solutions efficient. Thus in an efficient solution, any mixed batch processes either none or exactly a unit demand of an original vertex $v \in V$. Therefore, in an efficient solution, the demand $r(v)$ of each original vertex v, if covered only by mixed batches, is split into exactly $\deg(v)$ mixed batches.

We will assume that $S(G', r)$ is efficient. Let C be the set of original vertices whose demand is covered not only by mixed batches. Our claim is that C is a vertex cover. To see this, suppose, on the contrary, that both $r(u)$ and $r(v)$ are covered only by mixed batches. Then from the efficiency of $S(G', r)$, $r(u)$ and $r(v)$, respectively, are split into $\deg(u)$ and $\deg(v)$ batches, and therefore, there are two saturated mixed batches for the subdivision of uv, one processing a unit from $r(u)$ and all $r(u_e) = \lambda - 1$, and the other processing a unit of $r(v)$ and all $r(v_e) = \lambda - 1$. Therefore, for this subdivision, exactly three batches are required to cover the demands of u_e, m_e, and v_e; this contradicts to the proposal that $S(G', r)$ is efficient. Therefore, C is indeed a vertex cover of G. Therefore, the number of batches z_S of an efficient feasible solution $S(G', r)$ is

$$z_S \geq |C| + 2|E|. \tag{2}$$

Suppose the BCP can be approximated within a factor $(1+\epsilon)$ of the optimum. Let $S(G', r)$ be an approximate solution which may be assumed to be efficient. Then

$$\begin{aligned} \epsilon\mathrm{OPT}(f(I)) &\geq z_S - \mathrm{OPT}(f(I)) \\ &\geq |C| + 2|E| - \mathrm{OPT}(f(I)) \\ &\geq |C| - \mathrm{OPT}(I). \end{aligned} \tag{3}$$

The second and the third inequalities follow from (2) and (1), respectively. Because $|E| \leq \delta\mathrm{OPT}(I)$, (1) implies $\mathrm{OPT}(f(I)) \leq (2\delta + 1)\mathrm{OPT}(I)$ which, combined with (3), implies

$$|C| - \mathrm{OPT}(I) \leq \epsilon(2\delta + 1)\mathrm{OPT}(I).$$

Thus, an $(1+\epsilon)$-approximation algorithm of the batch consolidation problem implies an $(1 + \epsilon(2\delta + 1))$-approximation algorithm of VCB. However, when $\delta = 4$ VCB cannot be approximated within $\frac{53}{52}$ of the optimum unless P = NP [2]. Therefore, for any

$$\epsilon < \frac{1}{52} \cdot \frac{1}{9} \ (> 0.0021)$$

$(1+\epsilon)$-approximation is impossible for the batch consolidation problem. ∎

Remark 3.3. *The reduction used in the proof of Theorem 3.2 is the "*L*-reduction"* [11]*, and hence, the batch consolidation is Max-*SNP*-hard.*

4. A $\frac{3}{2}$-APPROXIMATION

For a problem (G, r) defined by $G = (V, E)$ and $r \in \mathbb{Q}_+^V$, the approximation algorithm is conveniently described by using an auxiliary problem (G_r, r')(Fig. 3) in which $V(G_r)$,

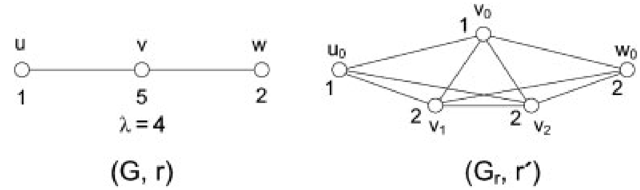

FIG. 3. Construction of auxiliary problem (G_r, r') from (G, r).

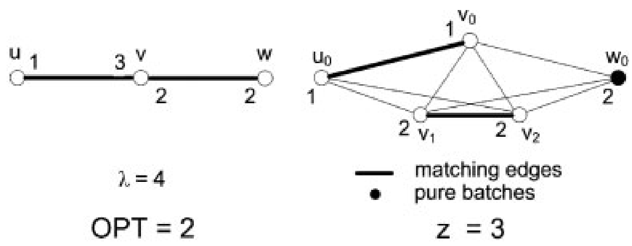

FIG. 4. Result of applying Algorithm 4.1 to the instance in Figure 3.

$E(G_r)$, and r' are defined as follows. For each $v \in V$ such that $r(v) > 0$, compute $k = \lceil r(v)/\lambda \rceil - 1$ and accordingly construct a set of $2k+1$ vertices, $V'_v = \{v_0, v_1, v_2, \ldots, v_{2k}\}$, their demands r' so that $r'(v_0) = r(v) - k\lambda$, $r'(v_1) = r'(v_2) = \cdots = r'(v_{2k}) = \lambda/2$, and an edge set $E'_v = \{xy | r'(x) + r'(y) \leq \lambda,\ x \neq y, \text{ and } x, y \in V'_v\}$. Also for each $uv \in E$, consider an edge set $E'_{uv} = \{xy | x \in V'_u,\ y \in V'_v, \text{ and } r'(x) + r'(y) \leq \lambda\}$. Finally, $V(G_r) = \bigcup_{v \in V} V'_v$ and $E(G_r) = (\bigcup_{v \in V} E'_v) \cup (\bigcup_{uv \in E} E'_{uv})$. For notational convenience, we allow vertices with zero demand.

Algorithm 4.1.

STEP 0. *As a preprocessing step, make the problem satisfy the condition of Proposition 2.2 by removing the smallest possible number of pure batches from each vertex; delete vertices with zero demand, if any.*

STEP 1. *Construct the auxiliary problem (G_r, r') of (G, r).*

STEP 2. *Compute a maximum matching M on G_r.*

STEP 3. *For each M-exposed vertex y of G_r, construct a pure batch processing $r'(y)$ of the demand of the corresponding original item. For each edge $xy \in M$, construct a mixed batch processing the demands $r'(x)$ and $r'(y)$ of the corresponding original items.*

STEP 4. *Return the $(|V(G_r)| - 2|M|)$ pure batches and $|M|$ mixed batches from Step 3 along with the pure batches from Step 0 as the solution of (G, r).*

Because the demands r' of the auxiliary problem are a decomposition of the original demands r, the batches from Step 3 are clearly a feasible solution of the original problem. Algorithm 4.1 applied to the instance of Figure 3 provides a solution (Fig. 4). $\text{OPT}(G, r)$ and $z(G, r)$ denote the numbers of batches of an optimal solution and a solution returned by Algorithm 4.1, respectively. G_r and r' themselves define a BCP and hence $z(G_r, r')$ is well-defined. Algorithm 4.1 applied to problem (G_r, r') returns the same solution $z(G_r, r') = z(G, r)$ when no pure batch is constructed from (G, r) by the preprocessing (Step 0).

We now define some more notation. For a partition $(S; T)$ of $V(G_r)$, r'_S and r'_T denote the r' restricted to S and T, respectively: $r'_S(x) = r'(x)\chi^S(x)$, $r'_T(x) = r'(x)\chi^T(x)$, $\forall x \in V(G_r)$.

Lemma 4.2. *Suppose the number of pure batches from Step 0 is zero. Then, for any partition $(S; T)$ of $V(G_r)$, $z(G, r) = z(G_r, r') \leq z(G_r, r'_S) + z(G_r, r'_T)$.*

Proof. If we apply Algorithm 4.1 to (G_r, r'_S) and (G_r, r'_T), then the two returned solutions correspond to maximum matchings M_1 of $G_r[S]$ and M_2 of $G_r[T]$. Then, $M_1 \cup M_2$ is a matching of G_r. Therefore, because no pure batch was obtained by Step 0,

$$\begin{aligned} z(G, r) &= |V(G_r)| - |M| \\ &\leq |V(G_r)| - (|M_1 \cup M_2|) \\ &= (|S| - |M_1|) + (|T| - |M_2|) \\ &= z(G_r, r'_S) + z(G_r, r'_T). \end{aligned}$$

∎

For a partition $(S; T)$ of $V(G_r)$, let σ and $\tau \in \mathbb{Q}_+^V$ be, respectively, the demands of r decomposed into S and T: $\sigma(v) = \sum_{x \in S \cap V'_v} r'_S(x)$, $\tau(v) = \sum_{x \in T \cap V'_v} r'_T(x)$, $\forall v \in V$. Note that $\sigma + \tau = r$. Also, V_S be the set of vertices $v \in V$ whose decomposition V'_v intersects with S: $V_S = \{v \in V | S \cap V'_v \neq \emptyset\}$. Also define V_T similarly.

Lemma 4.3. *If either, (i) $V_S \cap V_T = \emptyset$, (ii) $V_S \cap V_T = \{v\}$ and $\sigma(v) = k\lambda$ for some $k \in \mathbb{Z}_+$, or (iii) $V_S \cap V_T = \{v\}$ and $v_0 \in S$, then we have $z(G_r, r'_T) \leq z(G, \tau)$.*

Proof. For (i) or (ii), (G_r, r'_T) is the auxiliary problem of (G, τ) and therefore $z(G_r, r'_T) = z(G, \tau)$.

For (iii), suppose that T has q vertices associated with v (i.e., $T \cap V'_v = \{v_1, \ldots, v_q\}$). $v_0 \in S$, so $r'_T(v_1) = \cdots = r'_T(v_q) = \lambda/2$ and $\tau(v) = q \times \frac{\lambda}{2}$. If q is odd, then, by definition, v is decomposed into q vertices all of which have the demand $\lambda/2$ in the auxiliary problem (G_τ, τ') of (G, τ). This means that (G_r, r'_T) is the same as (G_τ, τ') as far as the nonzero-demand vertices are concerned. Hence, we have $z(G_r, r'_T) = z(G, \tau)$.

If q is even, then, in the auxiliary problem (G_τ, τ'), v is decomposed into $q-1$ vertices among which one vertex has the demand λ and the remaining $q-2$ vertices each has the demand $\lambda/2$. Thus, comparing (G_r, r'_T) and (G_τ, τ') over the vertices with nonzero demands, the only difference is that (G_τ, τ') has a vertex with the demand λ instead of the two vertices, say, v_{q-1} and v_q of the former both with the same demand $\frac{\lambda}{2}$. The vertex with the demand λ in (G_τ, τ') will be isolated, but the rest of vertices will, in their inducing subgraph, inherit the adjacency from (G_r, r'_T). Thus, if $M(G_\tau)$ is a maximum matching of G_τ, then $M = M(G_\tau) \cup \{v_{q-1}v_q\}$ is also a matching of (G_r, r'_T). Therefore

$$\begin{aligned} z(G_r, r'_T) &\leq |T| - |M| \\ &= (|V(G_\tau)| + 1) - (|M(G_\tau)| + 1) = z(G, \tau). \end{aligned}$$

∎

Lemma 4.4. *Let Δ be a constant such that $0 < \Delta \leq \lambda$. For any fixed $w \in V$, define s from r by increasing $r(w)$ by*

Δ: $s = r + \Delta\chi^w$. *Then,* $\mathrm{OPT}(G,r) \le \mathrm{OPT}(G,s) \le \mathrm{OPT}(G,r)+1$ *and* $z(G,s) \le z(G,r)+1$.

Proof. The first half of the statement is trivial. We prove the second half for each of the two subcases of $0 < r'(w_0) + \Delta \le 2\lambda$ in the auxiliary problem. It suffices to prove the lemma for the case when the number of pure batches from Step 0 is 0.

CASE 1. $r'(w_0)+\Delta \le \lambda$. Then, by definition of the auxiliary problem, $V(G_s) = V(G_r)$. Notice that no pure batch is constructed by Step 0 of Algorithm 4.1 when applied to (G,s). Also as, due to the increase in $r'(w_0)$, G_s has fewer edges incident to w_0 than does G_r, so their maximum matchings $M(G_s)$ and $M(G_r)$ from Step 2 satisfy $|M(G_s)| \ge |M(G_r)| - 1$. Therefore

$$z(G,s) = |V(G_s)| - |M(G_s)| \le |V(G_r)| - (|M(G_r)| - 1) = z(G,r)+1.$$

CASE 2. $\lambda < r'(w_0)+\Delta \le 2\lambda$. If $s(w) = r(w)+\Delta < (\deg(w)+1)\lambda$, then Step 0 returns no pure batch and there are $x, y \in V(G_s)$ such that $V(G_s) = V(G_r) \cup \{x,y\}$ and $s'(x) = s'(y) = \lambda/2$. $s'(w_0) = r'(w_0)+\Delta-\lambda \le r'(w_0)$ implies that all the edges incident to w_0 in G_r remain in G_s, so $M(G_r) \cup \{xy\}$ is a matching of G_s. Therefore

$$z(G,s) = |V(G_s)| - |M(G_s)| \le |V(G_r)|+2-(|M(G_r)|+1) = z(G,r)+1.$$

If $s(w) = r(w)+\Delta \ge (\deg(w)+1)\lambda$, Step 0 returns a single saturated pure batch and $V(G_s) = V(G_r)$. By the same argument as in the previous case, a maximum matching $M(G_r)$ of G_r is also a matching of G_s. Therefore

$$z(G,s) = |V(G_s)| - |M(G_s)| + 1 \le |V(G_r)| - |M(G_r)| + 1 = z(G,r)+1.$$

∎

Theorem 4.5. $z(G,r) \le \frac{3}{2}\mathrm{OPT}(G,r)$.

Proof. By induction on $\mathrm{OPT}(G,r)$. It suffices to prove the theorem for the case when the number of pure batches from Step 0 is 0.

Suppose $\mathrm{OPT}(G,r) = 1$. Then G is a graph of a single vertex or edge, in which cases, $z(G,r) = 1$. Assume that the theorem holds for any problem with OPT $< n$ and consider any problem (G,r) with $\mathrm{OPT}(G,r) = n$. Let $S(G,r)$ be any optimal solution. From Proposition 2.1 and 2.3, we may assume that $S(G,r)$ has at most one mixed batch for each edge of the solution graph and that the solution graph is a forest.

Suppose that $S(G,r)$ has a pure batch. Then, consider the demands q obtained by removing from r the demand processed by the pure batch. Let (G,q) denote the resulting problem. Then $\mathrm{OPT}(G,q) = n-1$ and from the induction hypothesis, $z(G,q) \le \frac{3}{2}\mathrm{OPT}(G,q)$. But, from Lemma 4.4, $z(G,r) \le z(G,q)+1$. Combining these two yields $z(G,r) \le \frac{3}{2}\mathrm{OPT}(G,q)+1 \le \frac{3}{2}\mathrm{OPT}(G,r)$.

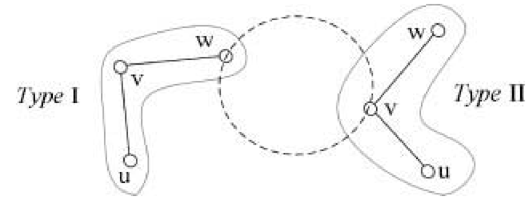

FIG. 5. Type I and II paths in the solution graph.

Suppose $S(G,r)$ has no pure batch. Then, its solution graph has no single-vertex component. If the graph has a single-edge component uv (which represents exactly one mixed batch), then remove from r the demands processed in the batch. Let (G,q) denote the resulting problem. Then, $\mathrm{OPT}(G,q) = n-1$ and the induction hypothesis implies $z(G,q) \le \frac{3}{2}\mathrm{OPT}(G,q)$. Applying Lemma 4.2 to the partition of $V(G_r)$ corresponding to the partition of V determined by the set $\{u,v\}$ and its complement in V yields $z(G,r) \le 1+z(G,q)$. Therefore, $z(G,r) \le \frac{3}{2}\mathrm{OPT}(G,q)+1 \le \frac{3}{2}\mathrm{OPT}(G,r)$.

Finally, suppose that $S(G,r)$ has neither a pure batch nor a single-edge component in its solution graph. Then, in the solution graph as a forest, a path $P = u-v-w$ of length 2 always exists such that $\deg(u) = 1$ and $\deg(v) = 2$(*type I*), or $\deg(u) = \deg(w) = 1$ and $\deg(v) \ge 3$(*type II*)(Fig. 5). Let $p(x)$ be the demands of x that are processed by the two mixed batches of P (hence $p(x) = 0$ if $x \in V \setminus V(P)$). For $x \in V$, define $q(x) = r(x) - p(x)$ and write the resulting problem as (G,q). Then, as $\mathrm{OPT}(G,q) = n-2$, the induction hypothesis implies $z(G,q) \le \frac{3}{2}\mathrm{OPT}(G,q)$.

Notice that to complete the proof, it suffices to find a partition $(S;T)$ of $V(G_r)$ satisfying the conditions of Lemma 4.2 and Lemma 4.3 and the relation

$$z\big(G_r, r'_S\big) + z(G,\tau) \le 3 + \frac{3}{2}\mathrm{OPT}(G,q). \qquad (4)$$

For, then we have $z(G,r) \le z(G_r,r'_S) + z(G_r,r'_T) \le z(G_r,r'_S) + z(G,\tau) \le 3 + \frac{3}{2}\mathrm{OPT}(G,q) = \frac{3}{2}\mathrm{OPT}(G,r)$ (where the first, second, and third inequalities follow from Lemma 4.2, Lemma 4.3, and (4), respectively). Thus, we will construct a partition $(S;T)$ of $V(G_r)$ satisfying the condition of Lemma 4.3 and (4) for each type of P. Note that $p(u), p(w) \le \lambda$ and $p(u)+p(v)+p(w) \le 2\lambda$ regardless of the type of P.

Suppose P is type I. Define $S = V'_u \cup V'_v$ and consider the corresponding partition of $(S;T)$ of $V(G_r)$. Then, $(S;T)$ satisfies the condition of Lemma 4.3: $V_S \cap V_T = \emptyset$. Because $r(u)+r(v) = p(u)+p(v) \le 2\lambda$ when P is type I, we can cover the demands r'_S with two batches and $z(G_r, r'_S) \le 2$. Finally, define σ and τ as in the discussion preceding Lemma 4.3. Then $\tau = q + p(w)\chi^w$. Because $p(w) \le \lambda$, τ satisfies the condition of Lemma 4.4 and $z(G,\tau) \le z(G,q)+1$. Therefore,

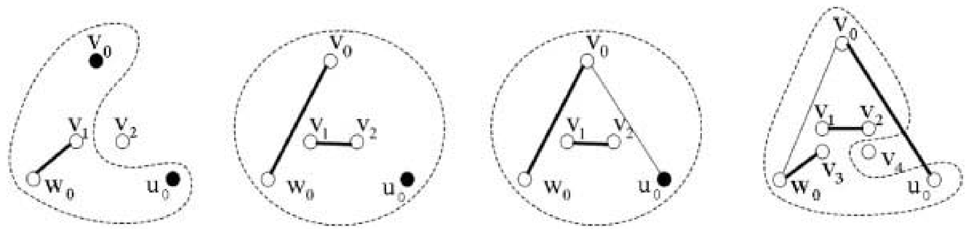

FIG. 6. The set S and the r'_S-covering batches for each subcase of Case 3 for Type II.

$$z(G_r, r'_S) + z(G,\tau) \le 2 + z(G,\tau) \le 3 + z(G,q) \le 3 + \frac{3}{2}OPT(G,q), \quad (5)$$

where the last inequality follows from the induction hypothesis on $OPT(G,q)$. Thus $(S;T)$ satisfies (4).

When P is type II, look at the following cases. When P is type II, $r'(u_0) = r(u) = p(u)$ and $r'(w_0) = r(w) = p(w)$.

Case 1. $p(v) \le \lambda$. Define $S = \{u_0, w_0\}$ and consider the corresponding partition of $(S;T)$ of $V(G_r)$. Then, $(S;T)$ satisfies the condition of Lemma 4.3: $V_S \cap V_T = \emptyset$. Also $z(G_r, r'_S) \le |S| = 2$. Finally, $\tau = q + p(v)\chi^v$. $p(v) \le \lambda$, so τ satisfies the condition of Lemma 4.4 and $z(G,\tau) \le z(G,q) + 1$. Therefore, (5) also holds in this case and $(S;T)$ satisfies (4).

Case 2. $\lambda < p(v)$ $(<2\lambda)$ and $p(u), p(w) \le \lambda/2$.

Let $S = \{u_0, v_1, v_2, w_0\}$. $V_S \cap V_T = \{v\}$ and $\sigma(v) = \lambda$, hence $(S;T)$ satisfies the condition of Lemma 4.3, and the demands r'_S can be covered by two batches $\{u_0v_1, v_2w_0\}$. Hence $z(G_r, r'_S) = 2$. Finally, $\tau = q + \Delta\chi^v$, where $\Delta = p(v) - \lambda (< \lambda)$, therefore (5) also holds in this case and $(S;T)$ satisfies (4).

Case 3. $\lambda < p(v)$ $(<2\lambda)$ and $p(u) > \lambda/2$. Then always $p(w) \le \lambda/2$.

We will construct a partition $(S;T)$ of $V(G_r)$ satisfying (i) the condition of Lemma 4.3 and (ii) $z(G_r, r'_S) \le 3$ and $\sigma \ge p$. Condition (ii) implies (4). Because $\tau = r - \sigma$ and $q = r - p$, $\tau \le q$ and hence $\mathrm{OPT}(G,\tau) \le \mathrm{OPT}(G,q) (= n-2)$ from Lemma 4.4. Applying the induction hypothesis to $\mathrm{OPT}(G,\tau)$ yields $z(G_r, r'_S) + z(G,\tau) \le 3 + \frac{3}{2}\mathrm{OPT}(G,\tau) \le 3 + \frac{3}{2}\mathrm{OPT}(G,q)$ and (4) follows.

We will consider the following three subcases: $\lambda < r'(v_0) + r'(w_0)$, $r'(v_0) + r'(w_0) \le \lambda < r'(u_0) + r'(v_0)$, and $r'(u_0) + r'(v_0) \le \lambda$. (The case assumptions imply $r'(v_0) + r'(w_0) < r'(u_0) + r'(v_0)$.)

Subcase 1. $\lambda < r'(v_0) + r'(w_0)$. Let $S = \{u_0, v_0, v_1, w_0\}$ (Fig. 6). Then, because $V_S \cap V_T = \{v\}$ and S contains v_0, $(S;T)$ satisfies the condition of Lemma 4.3.

Regarding $z(G_r, r'_S) \le 3$, choose the matching $\{v_1w_0\}$ to get the corresponding solution $\{u_0, v_0, v_1w_0\}$. Hence $z(G_r, r'_S) \le 3$.

Finally, for $\sigma \ge p$, it suffices to see $\sigma(v) \ge p(v)$, because $\sigma(u) = p(u)$ and $\sigma(w) = p(w)$. The case assumption $r'(u_0) > \lambda/2$, the subcase assumption $r'(v_0) + r'(w_0) > \lambda$, and $r'(v_1) = \lambda/2$ together imply that the total demand from $r'_S > 2\lambda$. But, then $\sigma(v) =$ the total demand from $r'_S - (\sigma(u) + \sigma(w)) > 2\lambda - (\sigma(u) + \sigma(w)) = 2\lambda - (p(u) + p(w)) \ge p(v)$, where the last inequality follows from the requirement that the total demand of $p \le 2\lambda$.

Subcase 2. $r'(v_0) + r'(w_0) \le \lambda < r'(u_0) + r'(v_0)$. Let $S = \{u_0, v_0, v_1, v_2, w_0\}$ (Fig. 6). If $r(v) \le 2\lambda$, then $V_S \cap V_T = \emptyset$. If $r(v) > 2\lambda$, then $V_S \cap V_T = \{v\}$ and $v_0 \in S$. Hence the condition of Lemma 4.3 is satisfied.

Choosing the matching $\{v_0w_0, v_1v_2\}$, we get $z(G_r, r'_S) \le 3$.

Finally, because $r'(v_1) = r'(v_2) = \lambda/2$ and $r'(u_0) + r'(v_0) > \lambda$, the total demand from $r'_S > 2\lambda$, which implies $\sigma \ge p$ by an argument similar to the one for Subcase 2.

Subcase 3. $r'(u_0) + r'(v_0) \le \lambda$.

Suppose that $r(v) \le 2\lambda$. Let $S = \{u_0, v_0, v_1, v_2, w_0\}$ (Fig. 6). Then, $V_S \cap V_T = \emptyset$ and the condition of Lemma 4.3 is satisfied. Choosing the matching $\{v_0w_0, v_1v_2\}$ yields $z(G_r, r'_S) \le 3$. Finally, because $\sigma(v) = r(v) \ge p(v)$, $\sigma \ge p$.

Suppose that $r(v) > 2\lambda$. Let $S = \{u_0, v_0, v_1, v_2, v_3, w_0\}$ (Fig. 6). Then, because $V_S \cap V_T = \{v\}$ and S contains v_0,

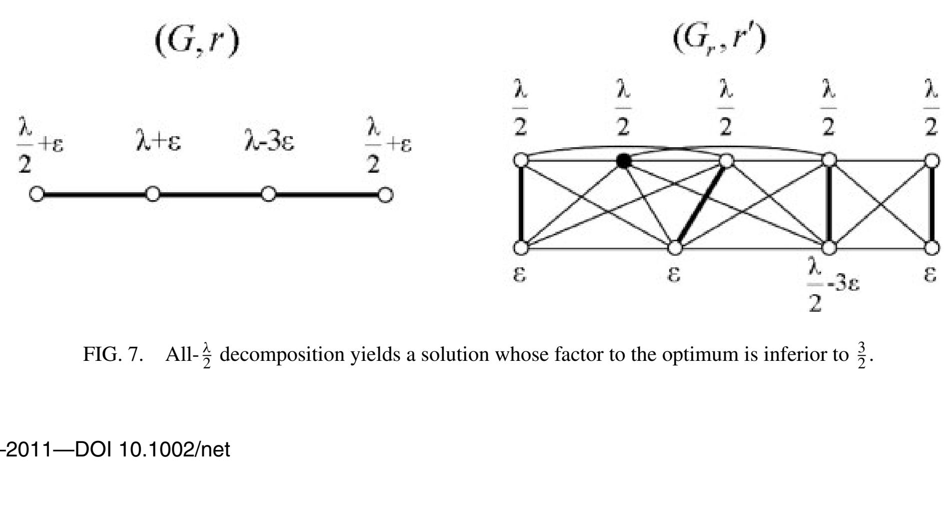

FIG. 7. All-$\frac{\lambda}{2}$ decomposition yields a solution whose factor to the optimum is inferior to $\frac{3}{2}$.

$(S;T)$ satisfies the condition of Lemma 4.3. Choosing the perfect matching $\{u_0v_0, v_1v_2, v_3w_0\}$ yields $z(G_r, r'_S) \leq 3$.

Finally, because $r'(v_1)+r'(v_2)+r'(v_3) = \frac{3}{2}\lambda$ and $r'(u_0) > \lambda/2$, the total demand from r'_S is $> 2\lambda$. Thus, by a similar argument, $\sigma \geq p$.

CASE 4. $\lambda < p(v)$ ($< 2\lambda$) and $p(w) > \lambda/2$. Then $p(u) \leq \lambda/2$ always holds.

Because of the symmetry of P in terms of u and w, the theorem also holds for this case from the argument obtained by interchanging u's and w's in the proof for Case 3.

Hence, the theorem follows. ∎

4.1. A Tight Example

Algorithm 4.1 applied to the instance of Figure 3 returns three batches, whereas an optimal solution uses two batches (Fig. 4). Thus, the approximation factor of Theorem 4.5 is realized.

4.2. Algorithm Based on All-$\frac{\lambda}{2}$ Decomposition

The proof of Theorem 4.5 involves the fact that $r'(v_0)$ may have a value up to λ in the decomposition of a vertex v. One might consider decomposing $r(v)$ into all $\lambda/2$- or less-sized demands in the auxiliary problem. However, such a decomposition may result in a solution whose objective value has the ratio $> \frac{3}{2}$ to the optimum.

Consider the instance(Fig. 7) in which the optimal number of batches is three. A possible maximum matching is indicated with thick edges in the auxiliary graph G_r based on the all-$\frac{\lambda}{2}$ decomposition. Then, the corresponding solution has five batches, which is inferior to a $\frac{3}{2}$-approximation. This solution cannot be improved easily by merging nonsaturated batches. Thus, the all-$\frac{\lambda}{2}$ decomposition is inferior to the proposed decomposition for the matching-based algorithm.

Remark 4.6. *Hong et al.* [5] *adopted the obvious analogy of the all-$\frac{\lambda}{2}$ decomposition, which may be referred to as all-$\frac{\lambda}{k}$ decomposition, in developing an approximation algorithm for the k-BCP for a fixed k that guarantees an approximation factor within twice that of the corresponding minimum cardinality k-set cover solution. Because the minimum cardinality* 2*-set cover problems is just a minimum cardinality matching problem which is solvable exactly, the k-batch algorithm provides a* 2*-approximation of the* 2*-BCP, which is inferior to our $\frac{3}{2}$-approximation. However,the decomposition scheme of our $\frac{3}{2}$-approximation does not strictly improve the approximation factor of their k-BCP approximation algorithm once k becomes as large as three* [5].

4.3. Complexity of the Approximation

The algorithm's complexity depends on the maximum matching algorithm implementation.

Proposition 4.7. *Algorithm 4.1 can be implemented in $O(|E|^{5/2})$ time.*

Proof. Step 0 can be performed in $O(|V|)$ time. In Step 1, the auxiliary problem can be constructed in $O(|E|^2)$ time because $|V(G_r)| \leq \sum_{v\in V} 2\lceil r(v)/\lambda\rceil \leq \sum_{v\in V} 2(\deg(v) + 1) = 4|E| + 2|V|$ and $|E(G_r)| = O(|V'|^2) = O(|E|^2)$. If we use the maximum matching implementation [10] on the auxiliary graph G_r, Step 2 requires $O(\sqrt{|V(G_r)|}|E(G_r)|) = O(|E|^{5/2})$ time, which determines the algorithm's complexity. ∎

5. FURTHER RESEARCH

One interesting study would be to explore the approximability of the generalization of the batch consolidation problem so that a single batch can process an arbitrary number of items. This problem is, as discussed in Section 1, more difficult than the clique partition problem. Also, a simpler proof of Theorem 4.5 would be valuable.

Acknowledgments

The authors are grateful to the anonymous referees for their helpful comments.

REFERENCES

[1] S. Arora and C. Lund, "Hardness of approximation," Approximation algorithms for NP-hard problems, D.S. Hochbaum (Editor), PWS, Boston, MA, 1997, pp. 399–446.

[2] M. Chlebik and J. Chlebikova, Complexity of approximating bounded variants of optimization problems, Theor Comput Sci 354 (2006), 320–338.

[3] S.Y. Chang, M.R. Chang, and Y.S. Hong, A lot grouping algorithm for a continuous slab caster in an integrated steel mill, Prod Planning Control 11 (2000), 363–368.

[4] L. Epstein and A. Levin, "On bin packing with conflicts," Proc 4th Workshop Approximation Online Algorithms, 2006, pp. 160–173.

[5] S.-P. Hong, M.-J. Park, and S.Y. Chang, Approximation of k-batch consolidation problem, Theor Comput Sci 410 (2009), 963–967.

[6] K. Jansen and Öhring. S., Approximation algorithms for time constrained scheduling, Inform Comput 132 (1997), 85–108.

[7] K. Jansen, An approximation scheme for bin packing with conflicts, J Combinatorial Optimizat 3 (1999), 363–377.

[8] K. Lee, S.Y. Chang, and Y.S. Hong, Continuous slab caster scheduling and interval graphs, Prod Planning Control 15 (2004), 495–501.

[9] C. Lund and M. Yannakakis, On the hardness of approximating minimization problems, J ACM 41 (1994), 960–981.

[10] S. Micali and V.V. Vazirani, "An $O(\sqrt{|V|}|E|)$ algorithm for finding maximum matching in general graphs," Proc 21st Ann Symp Foundations Comput Sci, Syracuse, New York, 1980, pp. 17–27.

[11] C.H. Papadimitiou and M. Yannakakis, Optimization, approximation and complexity classes, J Comput Syst Sci 43 (1991), 425–440.

[12] A. Schrijver, Combinatorial Optimization, Springer, Berlin, 2003.

[13] L. Tang, J. Liu, A. Ring, and Z. Yang, A review of planning and scheduling systems and methods for integrated steel production, Eur J Operat Res 133 (2001), 1–20.