IMA Journal of Mathematics in Management (1986) **1**, 1–11

Expected Behaviour of Line-Balancing Heuristics

ALAN FRIEZE

Department of Computer Science and Statistics, Queen Mary College, London

AND

ILKER BAYBARS

Graduate School of Industrial Administration, Carnegie–Mellon University, Pittsburgh, PA 15213

[Received October 1985 and in revised form February 1986]

We present a random model of assembly-line balancing problems and show that if edge density of the precedence network is low, then certain natural algorithms have asymptotically good performance in expectation and probable error.

1. Introduction

IN the standard assembly-line balancing (ALB) problem, we are given a finite set $V_n = \{1, 2, \ldots, n\}$ of tasks (together with the size x_i of each task), a partial order ρ (known as the 'precedence relations') defined on I, and a desired production rate $1/T$, and are asked to find an ordered partition $S = \{S_1, S_2, \ldots, S_m\}$ of I into nonempty sets such that

(a) $\sum_{i \in S_j} x_i \leq T$,

(b) if $i \rho j$ and $i \in S_h$ and $j \in S_k$, then $h \leq k$, and

(c) n is minimized subject to (a) and (b).

In ALB, x_i is the *processing time* of task i; and T is known as the *cycle time* and it is the length of time the item is available for processing at each station along the assembly line. If $i \rho j$, then the processing of task j cannot start until the processing of task i is completed. The subset S_k of I is the collection of tasks to be performed at the kth station along the assembly line. The objective in ALB is to minimize the idle time along the line which is equivalent to minimizing the number of stations (Salveson [27]).

If (b) above is dropped completely (i.e. if there are no precedence relations), the problem is the familiar bin-packing (BP) problem. ALB is a generalization of BP (Wee & Magazine [31]) in the sense that in BP, x_i is the weight or volume of item i and T is the capacity of a bin. In BP, the objective is to pack all the items in a minimum number of bins.

BP is NP-hard (e.g., Karp [20]) and hence so is ALB. Thus, unless P = NP, there cannot be a polynomially bounded exact algorithm for solving ALB. Hence, it is expected that determination of an optimal solution can require an excessive amount of computation. The computational studies of exact algorithms support this expectation (Johnson [17]). This drawback has necessitated the construction of inexact methods for finding good approximate solutions, in particular, fast heuristic algorithms (see Baybars [1] and Baybars [2], for surveys of exact and inexact ALB algorithms, respectively).

Whenever a heuristic method is used to solve a problem, a natural question arises as to the 'goodness' of the heuristic solution. Three analytical approaches are available to answer this question: (a) worst-case analysis, (b) probabilistic (expected behaviour) analysis, and (c) statistical analysis. Worst-case analysis establishes the maximum deviation from optimality; probabilistic analysis establishes the expected performance of the method or a bound on the probability that the heuristic solution is within a prespecified percentage of the optimal solution; and statistical analysis establishes the performance of the method by solving a large number of sample problems, thereby enabling one to draw some statistical inferences on the problem.

In Section 2 we describe some well-known bin-packing and line-balancing heuristics and summarize the known results on the worst-case analysis of these heuristics. In Section 3 we first present the known results on expected-behaviour analysis of bin-packing heuristics. We then initiate a similar analysis of line-balancing heuristics. Finally, in Section 4 we make some remarks and offer suggestions for future research.

2. Packing heuristics

All algorithms discussed here start with an ordered sequence of initially empty bins (stations) and the items (tasks) are packed (assigned) into (to) the bins (stations) one at a time. The input to BP algorithms can be described as a list $L=(x_1, x_2, \ldots, x_n)$ of item sizes. For ALB algorithms the input also includes a precedence relation ρ on $V_n = \{1, 2, \ldots, n\}$.

2.1 *Bin-Packing Heuristics*

We describe here four of the better-known algorithms. The first two algorithms do not sort (reorder) the items.

FF: for $i=1$ to n do
  pack x_i into the first bin in which it will fit od.

BF: for $i=1$ to n do
  pack x_i into the best bin in which it will fit (i.e. the one with least available space) od.

In FFD and BFD we sort L so that $x_1 \geqslant x_2 \geqslant \cdots \geqslant x_n$ and then apply FF and BF respectively. All four algorithms described above have the same order of time complexity: $O[n \log n]$. FF and BF are on-line algorithms, that is, packing is done in the order given without knowledge of the size and number of the later items.

2.2 *Line-Balancing Heuristics*

The algorithms described in the previous section can be applied to ALB problems by a minor modification to take into account the precedence relations.

The input L and ρ are first pre-processed so that L is in topological order (i.e., $i\rho j \rightarrow i<j$). We must be specific how this is done for our heuristics. Thus, let

$$\Gamma_\rho(i)=\{j\in V_n : i\rho j\} \quad (i\in V_n).$$

Also let $\{A_1, A_2, \ldots, A_m\}$ be the canonical partition of V_n defined by

$$A_1=\{i\in V_n:\Gamma_\rho(i)=\emptyset\}, \qquad A_k=\left\{i\in V_n\setminus\bigcup_{t=1}^{k-1}A_t \ : \ \Gamma_\rho(i)\subseteq\bigcup_{t=1}^{k-1}A_t\right\} \quad (k>1)$$

and m is the first index k for which, $V_n=\bigcup_{t=1}^{k}A_t$.

Now for $H\in$ HEURISTICS $=\{$FF, BF, FFD, BFD$\}$, we define a heuristic HP as follows:

Algorithm HP
  Input: list L, precedence relation ρ
begin
  Construct $A_1, A_2, \ldots, A_m$;
  Re-order L in the order $A_m, A_{m-1}, \ldots, A_1$;
  if $H\in\{$FFD, BFD$\}$ **then** sort tasks within each A_t into descending order;
  Let $(x_1, x_2, \ldots, x_n)$ now refer to the re-constructed list;
  for $i=1$ **to** n **do**
  Assign x_i to the first ($H\in\{$FF, FFD$\}$) or best ($H\in\{$BF, BFD$\}$) station, j say, into which it will fit, subject to all predecessors of x_i being already assigned to stations $1, 2, \ldots, j$ **od**
end

The main contribution of this paper is a probabilistic analysis of the performance of these heuristics.

2.3 *Worst-Case Analysis*

2.3.1 *Bin-Packing Heuristics* For input list L we let OPT(L) denote the number of bins used in an optimal solution. For heuristic H we let $H(L)$ denote the number of bins used by H. The following are well-known results (see, for instance, Johnson *et al.* [16]) for all L:

$$\mathrm{FF}(L)\le\tfrac{17}{10}\mathrm{OPT}(L)+2, \qquad \mathrm{BF}(L)\le\tfrac{17}{10}\mathrm{OPT}(L)+2,$$
$$\mathrm{FFD}(L)\le\tfrac{11}{9}\mathrm{OPT}(L)+4, \qquad \mathrm{BFD}(L)\le\tfrac{11}{9}\mathrm{OPT}(L)+4,$$

Hence, the asymptotic error bound of both FF and BF is 1·7 and they are 1·222 for FFD and BFD. Brown [5] has shown that no on-line algorithm can guarantee an asymptotic worst-case ratio better than 1·536. Thus, simple as it is, FF is a remarkably good algorithm.

Fernandez de la Vega & Lueker [10] and Karmarkar and Karp [19] have shown that the asymptotic worst-case ratio of 1 is achievable with a polynomial-time algorithm (that is, the number of excess bins is guaranteed to be 'little o' of OPT(L)).

2.3.2 *Line-Balancing Heuristics* Wee & Magazine [29, 30] have recently initiated the worst-case analysis of line-balancing heuristics. Before we discuss their results we extend the notation of Section 2.3.1: let $\text{OPT}(L, \rho)$ be the optimal (minimum) number of stations for some ALB problem and let $H(L, \rho)$ be the number of stations found using some heuristic method H for that given problem. Let T_j denote the work content of station j. Clearly, if the sum of the work contents of any pair of consecutive stations, as determined by the heuristic method, is less than the cycle time, then these two stations can be combined into one, thereby reducing the heuristic solution by 1. Based on this argument, Wee & Magazine have shown that if

$$T_j + T_{j+1} > T \quad \text{for } j = 1, 2, \ldots, H(L, \rho) - 1,$$

then $H(L, \rho) \leqslant 2\text{OPT}(L, \rho) - 1$. Following Wee & Magazine, a heuristic is *reasonable* if $T_j > T - x_{\max}$ for $j = 1, 2, \ldots, H(L, \rho) - 1$ where $x_{\max} = \max\{x_1, x_2, \ldots, x_n\}$. Wee & Magazine have shown that if H is reasonable, then

$$H(L, \rho) < \text{OPT}(L, \rho)/(1 - x_{\max}/T) + 1$$

thereby improving the first upper bound given above.

In a recent report, Queyranne [26] complements the Wee & Magazine results, assuming that $P \neq NP$. Queyranne has shown that there cannot exist a polynomial heuristic with worst-case ratio less than $\frac{3}{2}$ (thus no efficient heuristic method can guarantee a solution better than 150% of the optimal solution). However, if tasks are 'short', then the bounds can be improved slightly: if $x_{\max}$ is within ε of T (i.e., if $x_{\max} \leqslant \varepsilon T$), then the worst-case ratio is no more than $1/(1-\varepsilon)$ (Wee & Magazine [29]) but cannot be much better than $1/(1-\frac{1}{2}\varepsilon)$ for any polynomial-time heuristic algorithm. Wee & Magazine conjecture that the worst-case ratio of their algorithm IUFFD is $\frac{5}{3}$.

3. Expected behaviour analysis

3.1 *Expected Behaviour of Bin-Packing Heuristics*

The probabilistic model of bin packing is as follows: item sizes are independently and uniformly distributed in the interval $(0, u]$ for some $u \leqslant 1$ and the bins have unit capacities. Following Bently *et al.* [3], let L_n^u denote the random variable whose values are lists of n items generated according to this distribution. Also let $\Sigma(L)$ denote the lower bound $\sum_{x_i \in L} x_i \leqslant \text{OPT}(L)$. Then $\mathsf{E}\,\Sigma(L_n^u) = \frac{1}{2}un$.

We now summarize those results on the expected behaviour of bin packing heuristics that are needed in the sequel.

THEOREM 3.1 *The following are true:*

$$\mathsf{E}[\text{FF}(L_n^1) - \Sigma(L_n^1)] = O((n \log n)^{\frac{2}{3}}), \tag{3.1a}$$

$$\mathsf{E}[\text{BF}(L_n^1) - \Sigma(L_n^1)] = O(n^{\frac{1}{2}} \log n), \tag{3.1b}$$

$$\mathsf{E}[\text{FFD}(L_n^1) - \Sigma(L_n^1)] = O(n^{\frac{1}{2}}), \tag{3.1c}$$

$$\mathrm{E}[\mathrm{BFD}(L_n^1) - \Sigma(L_n^1)] = O(n^{\frac{1}{2}}), \tag{3.1d}$$

$$\mathrm{E}[\mathrm{FFD}(L_n^u) - \Sigma(L_n^u)] = O(1) \quad (u \leqslant \tfrac{1}{2}). \tag{3.1e}$$

The estimates (3.1a) and (3.1b) above are proved in Shor [28], and (3.1c) and (3.1d) are proved in Lueker [23] and (3.1e) is from [4].

For each heuristic $H \in$ HEURISTICS, there is a heuristic $H2$ which proceeds in the same manner as H, except that $H2$ never packs more than 2 items in a bin. We now give two lemmas:

LEMMA 3.2 *Let L' be a sub-list of L. Then;*

$$H2(L) \geqslant H(L) \quad \text{for all lists } L, \tag{3.2a}$$

$$H2(L') \leqslant H2(L) \quad \text{for all lists } L \text{ and } L' \text{ with } L' \subseteq L. \tag{3.2b}$$

Proof. The case $H = \mathrm{FF}$ was proved in [4] and the case $H = \mathrm{BF}$ was proved in [28]. The other 2 cases follow immediately. □

We now note the following:

LEMMA 3.3 *Replacing $H \in$ HEURISTICS by $H2$ in* (a)–(d) *of Theorem* 3.1 *yields the same conclusion.*

Proof. (a) and (b) are what is proved in Shor [28] and then he uses Lemma 2 to prove Theorem 1(a) and (b). Cases (c) and (d) follow from the work of Lueker [23]. He describes a bin-packing algorithm BINPACK1 for which he proves

$$\mathrm{E}\ \mathrm{BINPACK1}(L_n^1) = \tfrac{1}{2}n + O(n^{\frac{1}{2}}). \tag{3.3a}$$

But by removing the single sentence (Lueker [23: Part C of Proof of Thm 3])

> "Now if x_i is packed into a bin which already had 2 items S_{i-1} will be the same as S_i, so that the problem does not arise."

we obtain a proof of

$$\mathrm{BINPACK1}(L) \geqslant \mathrm{HD2}(L) \quad \text{for } H \in \{\mathrm{FF}, \mathrm{BF}\} \text{ and } L \text{ arbitrary.} \quad \square \tag{3.3b}$$

3.2 *Expected Behaviour of Line-Balancing Heuristics*

For our probabilistic model of ALB we have $x_1, x_2, \ldots, x_n$ generated as for BP. To generate a random precedence relation ρ we take $p = p(n)$ and then for each $i < j$ we independently include (i, j) in ρ with probability p and exclude it with probability $(1-p)$. The main results of this paper are summarized in the following theorem. It generalizes Theorem 3.1.

THEOREM 3.4 *Suppose $p(n) \to 0$ as $n \to \infty$. Then for $H \in$* HEURISTICS

(a)
$$\mathrm{E}\ H\mathrm{P}(L_n^1, \rho) = [1 + o(1)]\mathrm{E}\ \mathrm{OPT}(L_n^1, \rho), \tag{3.4a}$$

$$\Pr\{H\mathrm{P}(L_n^1, \rho) \geqslant [1 + o(1)]\mathrm{OPT}(L_n^1, \rho)\} = o(1), \tag{3.4b}$$

for suitable $o(1)$ terms.

(b) In particular assume $p = cn^{-\gamma}$ where $c, \gamma > 0$ are constants. Then, we have the following table:

H	$0<\gamma<1$	$\gamma=1$	$\gamma>1$
FF	$\Delta = O(n^{1-\gamma/3}(\log n)^{2\gamma/3})$	$\Delta = O(n^{2/3}(\log n))$	$\Delta = O((n \log n)^{2/3})$
BF	$\Delta = O(n^{1-\gamma/2}(\log n)^{\gamma})$	$\Delta = O(n^{1/2}(\log n)^{2/3})$	$\Delta = O(n^{1/2} \log n)$
FFD	$\Delta = O(n^{1-\gamma/2})$	$\Delta = O(n^{1/2}(\log n)^{1/2})$	$\Delta = O(n^{1/2})$
BFD	$\Delta = O(n^{1-\gamma/2})$	$\Delta = O(n^{1/2}(\log n)^{1/2})$	$\Delta = O(n^{1/2})$

where $\Delta = \Delta(H) = \mathsf{E}[\text{HP}(L_n^1, \rho) - \text{OPT}(L_n^1, \rho)]$.

We first prove two lemmas.

LEMMA 3.5 *Let* $H \in$ HEURISTICS *and let* L *and* ρ *be arbitrary. Let* $A_1, A_2, \ldots, A_m$ *be as defined prior to algorithm* HP. *Then,*

$$\text{HP}(L, \rho) \leq \sum_{i=1}^{m} \text{H2}(A_i). \tag{3.5a}$$

(Here $\text{H2}(A_i)$ *is the number of bins used by the* BP *algorithm* H2 *when applied to the list* A_i. *Although the* A_i *are defined as sets, we assume that the list* L *induces an ordering on them.)*

Proof. Let HP2 denote the algorithm that applies H2 separately to $A_m, A_{m-1}, \ldots, A_1$ and then concatenates its solutions in this order. Clearly

$$\text{HP2}(L, \rho) = \sum_{i=1}^{m} \text{H2}(A_i). \tag{3.5b}$$

We prove, by induction on $|L|$, the number of items in L, that

$$\text{HP}(L, \rho) \leq \text{HP2}(L, \rho). \tag{3.5c}$$

The base case $|L| = 1$ is trivial, and so assume that (3.5c) is true for all lists shorter than L. Let S and S' be the sets of items packed into the first bin by HP and HP2, respectively. It is clear that $S' \subseteq S$. Also, the remaining bins are packed as $\text{HP}(L \backslash S')$ and $\text{HP2}(L \backslash S')$, respectively. Thus;

$$\text{HP}(L) = 1 + \text{HP}(L \backslash S), \qquad \text{HP2}(L) = 1 + \text{HP2}(L \backslash S'). \tag{3.5, e}$$

We show that

$$\text{HP2}(L \backslash S') \geq \text{HP2}(L \backslash S). \tag{3.5f}$$

The induction hypothesis gives

$$\text{HP2}(L \backslash S) \geq \text{HP}(L \backslash S). \tag{3.5g}$$

Now, (3.5d)–(3.5g) together imply (3.5c). To prove (3.5f), let $B_i = A_i \backslash S$ for $i = 1, 2, \ldots, m$. We show that

$$u \in B_t \text{ and } t \neq 1 \text{ implies that there exists } v \in B_{t-1} \text{ such that } u \,\rho\, v. \tag{3.5h}$$

Suppose (3.5h) is not true for some $u \in B_t$. Then there exists a nonempty subset T

of S such that $T=\{v : v \in A_{t-1}$ and $u \rho v\}$. But then HP has packed T before packing u; a contradiction.

It follows from (3.5h) that if $B_t \neq \varnothing$ then $B_t, B_{t-1}, \ldots, B_1$ are all sets in the canonical partition of $L \backslash S$. Thus

$$\begin{aligned} \text{HP2}(L\backslash S) &= \sum_{i=1}^{m} \text{H2}(B_i) \\ &\leqslant \sum_{i-1}^{m-2} \text{H2}(A_i) + \text{H2}(A_{m-1}\backslash S') + \text{H2}(A_m \backslash S') \qquad \text{by (3.2b)} \\ &= \text{H2}(L\backslash S'). \quad \square \end{aligned}$$

LEMMA 3.6 *Suppose* $\varepsilon > 0$ *is such that* $p^{\varepsilon} \leqslant 1/\mathrm{e}$ (*where* e *is the base of natural logarithms*). *Suppose also that* H $\in$ HEURISTICS *is such that*

$$\mathsf{E}\, \text{H2}(L_n^1) \leqslant \tfrac{1}{2}n + an^{\alpha}$$

where $0<\alpha<1$ *and* $a>0$ *are constants. Then,*

$$\mathsf{E} \sum_{i=1}^{n} \text{H2}(A_i) \leqslant \tfrac{1}{2}n + a\left[np^{(1-\alpha)(1-\varepsilon)} + \left(\frac{n(ep^{\varepsilon})^{np^{1-\varepsilon}}}{p}\right)^{\alpha}\right] \qquad (3.6a)$$

Proof. Note that

$$\begin{aligned} \mathsf{E}\, \text{H2}(A_i) &= \sum_{k=0}^{n} \mathsf{E}[\text{H2}(A_i) \mid |A_i| = k] \Pr(|A_i| = k) \\ &\leqslant \sum_{k=0}^{n} (\tfrac{1}{2}k + ak^{\alpha}) \Pr(|A_i| = k) \\ &= \tfrac{1}{2}\mathsf{E}|A_i| + a\mathsf{E}|A_i|^{\alpha}. \end{aligned}$$

Thus, letting $A_i = \phi$ for $i > m$,

$$\begin{aligned} \sum_{i=1}^{n} \mathsf{E}\, \text{H2}(A_i) &\leqslant \tfrac{1}{2}\sum_{i=1}^{n} \mathsf{E}|A_i| + a \sum_{i=1}^{n} \mathsf{E}|A_i|^{\alpha} \\ &= \tfrac{1}{2}n + a\sum_{i=1}^{n} \mathsf{E}|A_i|^{\alpha} \\ &\leqslant \tfrac{1}{2}n + a \sum_{i=1}^{n} (\mathsf{E}|A_i|)^{\alpha} \qquad (3.6b) \end{aligned}$$

(by Jensen's inequality; see, for example, Feller [8]). Now $A_i \neq \varnothing$ implies that the digraph (V_n, ρ) contains a directed path of length $i-1$. Thus,

$$\begin{aligned} \Pr(A_i \neq \varnothing) &\leqslant \Pr(\text{there exists a path of length } i-1) \\ &\leqslant \mathsf{E}(\#\text{ of paths of length } i-1) = \binom{n}{i} p^{i-1}. \qquad (3.6c) \end{aligned}$$

Hence,

$$\mathsf{E}|A_i| \leqslant n \Pr(A_i \neq \varnothing) \leqslant n\binom{n}{i}p^{i-1}$$

Now, let $k=\lfloor np^{1-\varepsilon}\rfloor$. Then our assumption implies $k\geq 2np-1$. Hence,

$$\sum_{i=1}^{k}(\mathsf{E}|A_i|)^{\alpha}\leq k^{1-\alpha}\Big(\sum_{i=1}^{k}\mathsf{E}|A_i|\Big)^{\alpha} \leq k^{1-\alpha}n^{\alpha} \tag{3.6d}$$

and

$$\sum_{i=k+1}^{n}\mathsf{E}|A_i|^{\alpha}\leq\sum_{i=k+1}^{n}\left[n\binom{n}{i}p^{i-1}\right]^{\alpha}. \tag{3.6e}$$

Note that, for $i\geq k+1$, we have $\binom{n}{i+1}p^{i}\Big/\binom{n}{i}p^{i-1}\leq\frac{1}{2}$, and therefore the RHS of (3.6e) is bounded from above by

$$\left[n\binom{n}{k+1}p^{k}\right]^{\alpha}[1+(\tfrac{1}{2})^{\alpha}+(\tfrac{1}{4})^{\alpha}+(\tfrac{1}{8})^{\alpha}+\cdots]\leq\left[n\binom{n}{k+1}p^{k}\right]^{\alpha}\Big/(1-2^{-\alpha}).$$

Thus (3.6d) and (3.6e) imply

$$\begin{aligned}\sum_{i=1}^{n}(\mathsf{E}|A_i|)^{\alpha}&\leq k^{1-\alpha}n^{\alpha}+\left[n\binom{n}{k+1}p^{k}\right]^{\alpha}\Big/(1-2^{-\alpha})\\&\leq k^{1-\alpha}n^{\alpha}+\left[n\Big(\frac{nep}{k+1}\Big)^{k+1}\Big/p\right]^{\alpha}\Big/(1-2^{-\alpha}),\end{aligned}$$

using $\binom{n}{k+1}\leq\left(\frac{ne}{k+1}\right)^{k+1}$. The result now follows from (3.6b). □

Lemmas 3.5 and 3.6 show that the expected value of the solution produced by the various heuristics on (L_n^1,ρ) is bounded by the RHS of (3.6a).

Proof of Theorem 3.4. To prove (3.4a) we note first that if $p=o(1/n^2)$ then $\rho=\varnothing$ with probability going to 1 as n goes to ∞ and then the result follows from Theorem 3.1. Thus we can assume that $p\geq c/n^2$ for some constant $c>0$. In which case, putting $\varepsilon=\frac{2}{3}$ in (3.6a) yields (3.4a).

To prove (3.4b) let

$$A(L_n^1,\rho)=H(L_n^1,\rho)-\Sigma(L_n^1).$$

(3.4a) and $\mathsf{E}\,\Sigma(L_n^1)=\frac{1}{2}n$ implies that $\mathsf{E}\,A(L_n^1,\rho)=o(n)$ and hence $A(L_n^1)=o(n)$ a.s. by the Markov inequality. Since $\Sigma(L_n^1)\geq\frac{1}{2}[1-o(1)]n$ a.s. we have our result.

To prove part (b) of Theorem 3.4, we need only specify the values of ε to be used in (3.6a). We claim, and the reader can easily check, that the following suffice, in conjunction with Lemma 3.3:

$$\varepsilon=\begin{cases}2/(\gamma\log n) & \text{if } 0<\gamma<1,\\ \log\log n/\log n & \text{if } \gamma=1,\\ 1-\gamma^{-1}+\delta/\log n\quad(\delta\text{ large}) & \text{if } \gamma>1.\end{cases}$$

This completes the proof. □

We have not thus far used the rather unexpected result (3.1e). We would like to be able to say something about $\mathsf{E}\,\text{FFDP}(L_n^u, \rho)$. Lemma 3.5 is not much help here, since (3.1e) is specifically excluded in the statement of Lemma 3.3. We have to be satisfied, with a rather 'unnatural' algorithm, FFDP′, which simply applies FFD to the sets $A_1, A_2, \ldots, A_m$ independently and then concatenates the results as does *HP2* of Lemma 3.5. This yields the following:

THEOREM 3.7 *For* $0 < u \leq \frac{1}{2}$,

$$\mathsf{E}\,\text{FFDP}'(L_n^u, \rho) \leq \tfrac{1}{2}un + am_0 \tag{3.7a}$$

where $a > 0$ *is constant and* $m_0 = \min\left\{k : \binom{n}{k} p^{k-1} \leq 1/n\right\}$. *In particular, if* $p(n) = c/n^\alpha$, *then*

$$m_0 = \begin{cases} O(n^{1-\alpha}\log n) & \text{for } \alpha \leq 1, \\ O(1) & \text{for } \alpha > 1. \end{cases} \tag{3.7b}$$

Proof.

$$\mathsf{E}\,\text{FFDP}'(L_n^u, \rho) = \sum_{i=1}^{n} \mathsf{E}\,\text{FFD}(A_i), \leq \sum_{i=1}^{n} (\tfrac{1}{2}u\mathsf{E}|A_i| + a\mathsf{E}\,\delta A_i)$$

where $a > 0$ is constant and

$$\delta A = \begin{cases} 0 & \text{if } A = \varnothing, \\ 1 & \text{if } A \neq \varnothing. \end{cases}$$

This follows directly from (3.1e). Thus,

$$\mathsf{E}\,\text{FFDP}'(L_n^u, \rho) \leq \tfrac{1}{2}un + a\mathsf{E}\,m$$

where $m = \max\{i : A_i \neq \varnothing\}$ as before. It follows from (3.6c) that $\Pr(m \geq m_0) \leq 1/n$. Thus $\mathsf{E}\,m \leq m_0 - 1 + n\Pr(m \geq m_0) \leq m_0$ and (3.7a) follows. Equation (3.7b) follows by routine calculation. □

4. Remarks

The paper presents a random model of assembly-line balancing problems. We have shown that certain natural algorithms have asymptotically good performance in expectation and probable error, provided the edge density $p(n) \to 0$. We note that large practical problems tend to have a low edge density. From a theoretical point of view, it would be interesting to see if the algorithms are good for $p(n)$ constant.

Karp, Luby, & Marchetti-Spaccamela [21] have carried out a probabilistic analysis of multidimensional bin-packing problems. We hope to generalize their results to assembly-line balancing with resource constraints.

REFERENCES

[1] BAYBARS, I. A survey of exact algorithms for the simple assembly line balancing problem. *Management Sci.* (in the press).

[2] BAYBARS, I., A survey of inexact algorithms for the simple assembly line balancing problem. WP #86-82-83, GSIA, Carnegie-Mellon University, Pittsburgh, PA (September 1984).
[3] BENTLEY, J. L., JOHNSON, D. S., LEIGHTON, T., & MCGEOCH, C. C. An experimental study of bin packing. *Proc. 21st Ann. Alleton Conf. on Communication, Control and Computing,* University of Illinois, Urbana, Ill (1983) 51–60.
[4] BENTLEY, J. L., JOHNSON, D. S., LEIGHTON, F. T., MCGEOCH, G. C., & MCGEOCH, L. A. Some unexpected behavior results for bin packing. *Proc. 16th ACM Symposium on Theory of Computing* (1984) 279–288.
[5] BROWN, D. J. A lower bound for on-line one-dimensional bin packing algorithms. R-864 CSL, University of Illinois, Urbana, Ill. (1979).
[6] COFFMAN, E. G. *et al.* Bin packing: maximizing the number of pieces packed. *Acta Informatica* **9** (1978) 263–271.
[7] COFFMAN, E. G., JR., SO, K., HOFRI, M., & YAO, A. C. A stochastic model of bin-packing. *Infor. and Control* **44** (1980) 105–115.
[8] COFFMAN, E. G., JR., GAREY, M. R., & JOHNSON, D. S. Approximation algorithms for bin-packing—An updated survey. To appear.
[9] FELLER, W. *An Introduction to Probability Theory and Its Applications,* Volumes I and II, Wiley, New York (1966) and (1968), respectively.
[10] FERNANDEZ DE LA VEGA, W., & LUEKER, G. S. Bin packing can be solved within $1+\varepsilon$ in linear time, *Combinatorica* **1** (1981) 312–320.
[11] FREDERICKSON, G. N. Probabilistic analysis for simple one- and two-dimensional bin packing algorithms. *Information Processing Letters* **11** (1980) 156–161.
[12] GAREY, M. R., GRAHAM, R. L., JOHNSON, D. S., & YAO, C. C. Resource constrained scheduling as generalized bin packing. *J. combinatorial Theor. Ser. A* **21** (1976) 257–298.
[13] GAREY, M. R., & JOHNSON, D. S. Approximation algorithms for bin packing problems—A Survey. *Analysis and Design of Algorithms in Combinatorial Optimization* (G. Ausiello & M. Lucertini, Eds). Springer-Verlag, New York (1981) 147–172.
[14] JOHNSON, D. C. Near-Optimal Bin Packing Algorithms. Doctoral Dissertation, Department of Mathematics, MIT, Cambridge, MA. (1973).
[15] JOHNSON, D. S. Fast algorithms for bin packing. *J. Computer Syst. Sci.* **8** (1974) 272–314.
[16] JOHNSON, D. S., DEMERS, A., ULLMAN, J. D., GAREY, M. R., & GRAHAM, R. L. Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM J. Comput.* **3** (1974) 299–325.
[17] JOHNSON, R. V. Assembly line balancing algorithms: Computation comparisons. *IJPR* **19** (1981) 277–287.
[18] KARMARKAR, N. Probabilistic analysis of some bin-packing algorithms. *Proc. 23rd Ann. Symp. on Foundations of Computer Science.* IEEE Computer Society, Los Angeles, CA (1982) 107–111.
[19] KARMARKAR, N., & KARP, R. M. An efficient approximation scheme for the one-dimensional bin packing problem. *Proc. 23rd Ann. Symp. on Foundations of Computer Science.* IEEE Computer Society, Los Angeles, CA (1982) 312–320.
[20] KARP, R. M. Reducibility among combinatorial problems, In *Complexity of Computer Computations* (R. E. Miller & J. W. Thatcher, Eds). Plenum Press, New York (1972) 85–104.
[21] KARP, R. M., LUBY, M., & MARCHETTI-SPACCAMELA, A. Probabilistic analysis of multidimensional bin packing problems. *Proc. 16th ACM Symp. on theory of Computing* (1984) 289–298.
[22] KNODEL, W. A bin packing algorithm with complexity $O(n \log n)$ and Performance 1 in the Stochastic Limit. *Conference of Mathematical Foundations in Computer Science,* Strbske Pleso, Czechoslovakia, (1981).
[23] LUEKER, G. S., An average-case analysis of bin packing with uniformly distributed

item sizes. Report No. 181, Department of Information and Computer Science. University of California, Irvine, CA (1982).

[24] LUEKER, G. S. Bin packing with items uniformly distributed over intervals [a, b]. *Proc. 24th Ann. Symp. on Foundations of Computer Science,* IEEE Computer Society, Los Angeles, CA (1983) 289–297.

[25] ONG, H. L., MAGAZINE, M. J., & WEE, T. S. Probabilistic analysis of bin packing heuristics. *Operations Res.* **32** (1984) 983–998.

[26] QUEYRANNE, M. Bounds on assembly line balancing heuristics. *Operations Res.* (to appear).

[27] SALVESON, M. E. The assembly line balancing problem. *J. Indust. Eng.* **6** (1955) 18–25.

[28] SHOR, P. The Average Case Analysis of Some On-Line Algorithms for Bin Packing. *Proc. 25th Ann. Symposium on Foundations of Computer Science, IEEE* (1984) 193–200.

[29] WEE, T. S., & MAGAZINE, M. J. An efficient branch and bound algorithm for assembly line balancing—Part I: Minimize the number of work stations. University of Waterloo, Working Paper (1981).

[30] WEE, T. S., & MAGAZINE, M. J. An efficient branch and bound algorithm for assembly line balancing—Part II: Maximize the production rate. University of Waterloo, Working Paper (1981).

[31] WEE, T. S., & MAGAZINE, M. J. Assembly line balancing as generalized bin packing, *Operations Res. Lett.* **1** (1982) 56–59.

[32] YAO, A. C. New algorithms for bin packing. *ACM* **27** (1980) 207–227.