J. Inform. Process. Cybernet. **EIK 27** (1991) 5/6, 263 – 271
(formerly: Elektron. Inform.verarb. Kybernet.)

A Three-dimensional Bin Packing Algorithm [1])

By *Guntram Scheithauer*

Abstract: In this paper an approximation algorithm for the three-dimensional bin packing problem is proposed and its performance bound is investigated. To obtain such a bound a modified bin packing algorithm is considered for a two-dimensional problem with bounded bin and its area utilization is estimated. Finally, a hard example gives a lower bound of the performance bound.

1. Introduction

In the last twenty years the two-dimensional bin packing problem was investigated by several authors and many papers were published on approximation algorithms, performance bounds, asymptotic behavior and further problems connected with the two-dimensional bin packing problem (see e.g. [1] – [4]). The two-dimensional bin packing problem is NP-hard ([2]). For this reason, especially, numerous approximation algorithms have been developed and investigated with respect to their performance behavior.

In this paper we consider the three-dimensional bin packing problem as a generalization of the two-dimensional one. This problem is the following:

Given a container with fixed length and width and unbounded height and a set of (rectangular) pieces. Find an orthogonal packing of the pieces within the container realizing a minimal required container height.

For this problem we propose an approximation algorithm which fills the container according to a "layer"-strategy, i.e., at first pieces are located on the bottom of the container, if no more piece can be fitted then the layer is closed and now location takes place on the top of the layer, and so on.

Further on, we investigate the performance bound of the approximation algorithm, i.e. the ratio of the heights used by the approximation algorithm and an optimal algorithm. To do so, we have at first to study the two-dimensional problem where, additionally, the bin width is bounded. Especially, we are interested in statements about the area utilization of approximation algorithms. This problem has not been considered so far in the literature.

The paper is organized as follows. In Section 2 we describe the NFD algorithm used for the bounded bin. After that, in Section 3, we investigate the area utilization of this algorithm and give further trivial bounds. A general three-dimensional approximation bin packing algorithm is proposed in Section 4 followed by investigations of its performance bound. In Section 6 the NFD algorithm is applied within the three-dimensional bin packing algorithm and corresponding performance bounds are shown. At last, a hard example gives a lower bound for the performance behavior of the proposed algorithm.

[1]) Keywords: cutting stock problem, bin packing problem, approximation algorithms, performance bounds

2. The NFD algorithm for bounded bins

Throughout the paper we assume that the given rectangles (pieces) are oriented each having a specified side that must be parallel to the bottom (length side) of the bin. With no loss of generality, let the length L and width W of the bounded bin be normalized to 1.

Further on, we assume that the set of rectangular pieces T_i, having length l_i and width w_i, $i = 1, \ldots, m$, fulfill the following conditions:

$$\sum_{i=1}^{m} l_i w_i \geqq 1\,, \quad \text{("sufficient many pieces")} \tag{1}$$

$$0 < l_i \leqq 1\,, \qquad 0 < w_i \leqq 1\,, \qquad i = 1, \ldots, m\,, \tag{2}$$

$$w_1 \geqq w_2 \geqq \ldots \geqq w_m\,. \tag{3}$$

For abbreviation, let be

$$\underline{l} = \min_{1 \leqq i \leqq m} l_i\,, \qquad \bar{l} = \max_{1 \leqq i \leqq m} l_i\,, \qquad \underline{w} = w_m = \min_{1 \leqq i \leqq m} w_i\,,$$
$$\bar{w} = w_1 = \max_{1 \leqq i \leqq m} w_i\,.$$

We use the NFDW (next fit decreasing width) algorithm as defined in [2]. The NFDW algorithm assumes a list L of rectangles ordered according to (3) and packs the rectangles in the order given by L so as to form a sequence of levels. All rectangles will be placed with their bottoms (length sides) resting on one of these levels. The first level is simply the bottom of the bin. Each subsequent level is defined by a horizontal line drawn through the top of the first (and hence maximum width) rectangle placed on the previous level.

With the NFDW algorithm, rectangles are packed left-justified on a level until there is insufficient space at the right to accommodate the next rectangle. At that point, the new level is defined, packing on the current level is discontinued, and packing proceeds on the new level. This packing process ends in the original (unbounded) problem if all pieces are packed. In our problem the packing is finished if there is insufficient space between the new level and the top of the bin. (This modified NFDW algorithm will be called NFD algorithm in the following.)

The space between two consecutive levels is called a *block*. Hence, packings may be regarded as a sequence of blocks $B_1, B_2, \ldots, B_t$, where the index increases from the bottom to the top of the packing. Let A_k denote the total area of the rectangles in block B_k, and let W_k denote the width of block B_k. Note that, by the manner in which the algorithm defines levels, we have $W_1 \geqq W_2 \geqq \ldots \geqq W_t$.

NFD algorithm

S0: $k := 1$, $\pi_1 := 1$, $W_1 := w_1$.

S1: Compute the maximal index j ($\leqq m$) with $\sum_{i=\pi_k}^{j} l_i \leqq 1$.

S2: Set $k := k + 1$, $\pi_k := j + 1$.

S3: If $\pi_k > m$ then go to S6.

S4: Set $W_k := w_{\pi_k}$.

S5: If $\sum_{j=1}^{k} W_j \leqq 1$ then go to S1.

S6: Set $t := k - 1$, $\pi_{t+1} := j$.

(π_j denotes the index of the first piece placed in block B_j, $j = 1, \ldots, t$; π_{t+1} is the index of the last piece packed.)

3. Estimations of the area utilization

Consider the NFD packing of such a list L, with blocks $B_1, B_2, \ldots, B_t$. For each k, let y_k be the total length of rectangles in B_k. Then we have:

$$y_k = \sum_{i=\pi_k}^{\pi_{k+1}-1} l_i > 1 - \bar{l},$$

since otherwise an additional piece could be packed in the block B_k. Because of (3) the used area A_k in block B_k can be estimated by

$$A_k \geqq y_k \cdot W_{k+1} > (1 - \bar{l}) \cdot W_{k+1}. \tag{4}$$

Formula (4) is valid for $k = 1, \ldots, t - 1$. For the last block B_t we have:

$$A_t > y_t \cdot \left(1 - \sum_{k=1}^{t} W_k\right) > (1 - \bar{l}) \cdot \left(1 - \sum_{k-1}^{t} W_k\right). \tag{5}$$

Let A denote the total used area in the bin. Summing up (4) and (5) we obtain:

$$\begin{aligned} A &= \sum_{k=1}^{t} A_k \\ &> \sum_{k=1}^{t-1} (1 - \bar{l}) \cdot W_{k+1} + (1 - \bar{l}) \cdot \left(1 - \sum_{k=1}^{t} W_k\right) \\ &= (1 - \bar{l}) \cdot (1 - W_1). \end{aligned}$$

Hence,

$$A > (1 - \bar{l}) \cdot (1 - \bar{w}) =: \beta(\text{NFD}). \tag{6}$$

It is to remark that β(NFD) depends on the range of the input data.

The following example shows that the estimation (6) is asymptotically sharp.

Example 1. Let be given positive integers r and s and let $\varepsilon < \min\{1/r, 1/s\}$. For abbreviation, we set $l = 1/r$ and $w = 1/s$. The (infinite) list L of rectangles T_i, $i = 1, 2, \ldots,$ is defined as follows:

$$T_i \text{ with } \begin{cases} l_i = \varepsilon, & w_i = w \quad \text{if} \quad i = 1, \\ l_i = \varepsilon, & w_i = \varepsilon \quad \text{if} \quad i > r \quad \text{and} \quad i \bmod r = 0, \\ l_i = l, & w_i = \varepsilon \quad \text{otherwise}. \end{cases}$$

Hence, the assumptions (1), (2) and (3) are fulfilled with $\bar{l} = l$, $\bar{w} = w$, $\underline{l} = \underline{w} = \varepsilon$. The NFD algorithm generates the following blocks:

B_1 contains the pieces $T_1, T_2, \ldots, T_r$, with $y_1 = \varepsilon + (r-1)\,l$,
B_k contains the pieces $T_{(k-1)r+1}, \ldots, T_{kr}$, with $y_k = y_1$,
$k = 2, 3, \ldots, [(1-w)/\varepsilon] + 1$.

(Here and in the following, $[a]$ denotes the largest integer not greater than a.) Hence,

$$\begin{aligned} A &= w \cdot \varepsilon + (r-1) \cdot l \cdot \varepsilon + [(1-w)/\varepsilon] \cdot \varepsilon \cdot ((r-1)\,l + \varepsilon) \\ &= \frac{r-1}{r} \cdot (\varepsilon + 1 - w) + O(\varepsilon) \\ &= (1 - \bar{l}) \cdot (1 - \bar{w}) + O(\varepsilon). \end{aligned}$$

If ε tends to zero then A tends to $\beta(\text{NFD})$. Hence, (6) yields an asymptotically sharp bound.

On the other hand, (6) is unsuitable if $\bar{l}$ or $\bar{w}$ tends to 1. For that reason, we consider a trivial bound for the area utilization. Obviously, at least $[1/\bar{l}] \cdot [1/\bar{w}]$ pieces, each having an area of at least $\underline{l} \cdot \underline{w}$ can be allocated in the unit square. Hence, an admissible two-dimensional packing algorithm ALG should fulfill the following demand:

$$\beta(\text{ALG}) \geqq [1/\bar{l}] \cdot [1/\bar{w}] \cdot \underline{l} \cdot \underline{w} =: \beta_0 \tag{7}$$

where $\beta(\text{ALG})$ denotes the area utilization using algorithm ALG.

Remark. If $\underline{l}$ tends to $\bar{l}$ and $\underline{w}$ tends to $\bar{w}$ (pieces of "equal size") or if $\bar{l}$ and $\bar{w}$ tend to 0 with $\bar{l}/\underline{l} \leqq$ const and $\bar{w}/\underline{w} \leqq$ const (relative "small pieces") then the utilization of the area of the unit square tends to 1.

In Example 1, β_0 tends to zero if ε tends to zero.

Example 2. Let

$$\underline{l} = 1/11 \leqq l_i \leqq 1/10 = \bar{l}, \quad i = 1, \ldots, m,$$
$$\underline{w} = 1/3 \leqq w_i \leqq 1/2 = \bar{w}, \quad i = 1, \ldots, m.$$

Then, $\beta(\text{NFD}) = 0.45$ and $\beta_0 = 0.6060\ldots$

In this example, (7) yields a better result than (6) does. One reason for this insufficiency is the fact that all estimations, done for getting (6), correspond to the unused length and width, not to the used area. For that reason, more detailed investigations are necessary.

Summarizing we have, the area utilization of the NFD algorithm is not worse than $\max\{\beta(\text{NFD}), \beta_0\}$.

4. A three-dimensional approximation bin packing algorithm

Now we will consider the three-dimensional packing problem. Let be given a container with length L, width W and unbounded height H and a set of (rectangular) pieces T_i having length l_i, width w_i and height h_i, $i = 1, \ldots, m$. Find an orthogonal packing of the m pieces within the container realizing a minimal required container height.

Without loss of generality we assume

$$L = 1, \quad W = 1, \quad 0 < l_i \leqq 1, \quad 0 < w_i \leqq 1, \quad i = 1, \ldots, m, \tag{8}$$

$$h_1 = 1 \geqq h_2 \geqq \ldots \geqq h_m > 0. \tag{9}$$

The approximation algorithm works as follows: The height h_1 of piece T_1 defines the height H_1 of the first layer (block) in the container parallel to the bottom side. Then, as a two-dimensional packing problem, the unit square $L \times W$ is filled with the bottom rectangles $l_i \times w_i$ of the pieces T_1, T_2, ... (according to a chosen two-dimensional packing algorithm A) until all pieces are packed or the next piece in the sequence, say T_{π_2}, cannot be packed within the unit square. If not all pieces are packed then the layer is closed and the next layer having height h_{π_2} is defined, and so on. If all pieces are packed then the packing process is finished. The required container height H_A (it depends on the algorithm A) equals the sum of the heights of the defined layers. More formally, we have:

3DBP algorithm

S0: $k := 1$, $\pi_1 := 1$.

S1: If $\pi_k > m$ then set $t := k - 1$ – stop.

S2: Compute the maximal index $j\,(\leqq m)$ with $\sum_{i=\pi_k}^{j} l_i w_i \leqq 1$.

Set $r := \pi_k$, $s := j$.

S3: Let $I_k := \{i : \pi_k \leqq i \leqq j\}$.
Use the two-dimensional packing algorithm A to allocate (pack) the rectangles $l_i \times w_i$, $i \in I_k$, within the unit square.

S4: If not all pieces of I_k are allocated then set

$$s := j, \qquad j := [(r + j)/2],$$

if $j > r$ then go to S3
else go to S6.

S5: If all pieces of I_k are allocated then set

$$r := j, \qquad j := [(j + s + 1)/2],$$

if $j < s$ then go to S3.

S6: A new layer is obtained: set

$$H_k := h_{\pi_k}, \qquad k := k + 1, \qquad \pi_k := r + 1,$$

go to S1.

Remarks.

In step 3 of the algorithm the pieces are allocated with respect to the assumed pre-ordering (9). Thereby it is essential that the remaining unpacked pieces have heights not greater than the heights of the packed pieces.

The two-dimensional packing algorithm A used in step 3 of the 3DBP algorithm may be a (two-dimensional) bin packing algorithm by itself or a more general two-dimensional packing algorithm.

To calculate the expense of the 3DBP algorithm we denote with $\alpha(\mathrm{A})$ the computational expense of algorithm A to allocate m pieces in a strip of length 1 (worst case bound of algorithm A). The bisection strategy defined by the steps 3, 4 and 5 yields a multiplicative factor of at most $\ln(m)$ of computational effort. The number of layers is at most m. Hence, the total expense of the 3DBP algorithm is bounded by $O(m \cdot \ln(m) \cdot \alpha(\mathrm{A}))$. If the algorithm A is polynomial then the 3DBP algorithm is, too.

5. Performance bounds

Because of $L \cdot W = 1$ the total volume of all pieces is a lower bound of the optimal height H_{opt} used by the best possible packing algorithm, i.e.

$$H_{\mathrm{opt}} \geqq \sum_{i=1}^{m} h_i l_i w_i .$$

Further on,

$$\sum_{i=1}^{m} h_i l_i w_i = \sum_{k=1}^{t-1} \sum_{i=\pi_k}^{\pi_{k+1}-1} h_i l_i w_i + \sum_{i=\pi_t}^{m} h_i l_i w_i \geqq \sum_{k=1}^{t-1} H_{k+1} \sum_{i=\pi_k}^{\pi_{k+1}-1} l_i w_i .$$

Let $\beta(\mathrm{A})$ denote the (lower bound of) utilization of the unit square by the algorithm A in each layer, then we have:

$$H_{\mathrm{opt}} \geqq \sum_{k=1}^{t-1} H_{k+1} \cdot \beta(\mathrm{A}) .$$

Since $H_1 = 1$, it follows:

$$\frac{1}{\beta(\mathrm{A})} H_{\mathrm{opt}} + 1 \geqq \sum_{k=1}^{t} H_k = H_{\mathrm{A}} \tag{10}$$

Remarks.

Depending on the knowledge of a given algorithm A, (10) gives a relative performance bound for the 3DBP algorithm.

The larger $\beta(\mathrm{A})$ is, the better is the estimation for the 3DBP algorithm.

6. Applying the NFD algorithm within the 3DBP algorithm

To apply the NFD algorithm in step 3 of the 3DBP algorithm we have to sort the pieces in I_k with respect to decreasing widths. (This implies a total computational effort of $O(m \cdot \ln(m) \cdot (m \cdot \ln(m) + \alpha(\mathrm{NFD})))$. Using an initial ordering of the pieces with respect to their widths and because of $\alpha(\mathrm{NFD}) = O(m)$ the total computational expense equals $O(m^2 \ln(m))$.)

Using (6) and (7) we have

$$H_{\mathrm{NFD}} \leqq \varrho \cdot H_{\mathrm{opt}} + 1 \tag{11}$$

with

$$\varrho = \varrho(\underline{l}, \underline{w}, \bar{l}, \bar{w}) = \min\left\{\frac{1}{\underline{l} \cdot \underline{w} \cdot [1/\bar{l}] \cdot [1/\bar{w}]}, \frac{1}{(1-\bar{l}) \cdot (1-\bar{w})}\right\}.$$

The performance bound given in (11) depends on the range of the input data. Let be $\mu = \min\{\underline{l}, \underline{w}\}$ and $\mathcal{M} = \max\{\bar{l}, \bar{w}\}$.

Remarks.

If μ tends to 0 and $\mathcal{M}$ tends to 1 then ϱ tends to infinity and (11) gives an unsuitable bound. On the other hand, we have

$$H_{\mathrm{NFD}} \leqq \frac{1}{(1-\mathcal{M})^2} \cdot H_{\mathrm{opt}} + 1$$

which implies $\varrho \leqq 4$ if $\mathcal{M} \leqq \frac{1}{2}$, and ϱ tends to 1 if $\mathcal{M}$ tends to 0, i.e. the proposed 3DBP algorithm is asymptotically exact.

If we additionally assume $\mu \geqq \delta \cdot \mathcal{M}$ with $\delta \in (0, 1)$ then $\varrho \leqq \bar{\varrho}(\mathcal{M})$ with

$$\bar{\varrho}(\mathcal{M}) = \min\left\{\frac{1}{\delta^2 \mathcal{M}^2[1/\mathcal{M}]^2}, \frac{1}{(1-\mathcal{M})^2}\right\}. \qquad \mathcal{M} \in (0, 1)$$

(see Fig. 1).

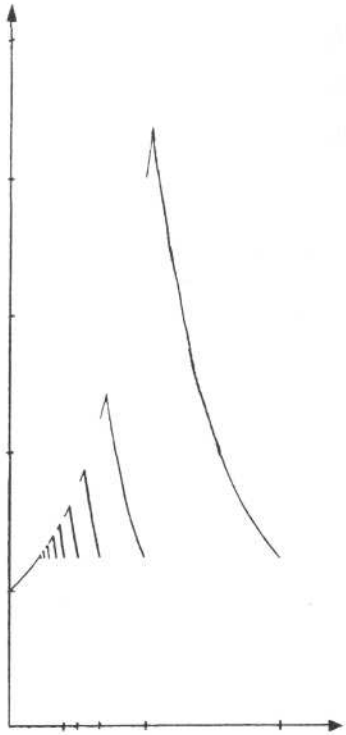

Fig. 1. $\bar{\varrho}(\mathcal{M})$ with $\delta = 0.9$

7. A lower bound for the 3DBP algorithm

Let us consider the following

Example 3.

$L = W = 1$, $\varepsilon < 1/6$, $m \geqq 4$, $m \bmod 4 = 0$,

m pieces of type T_a: $l_i = \frac{1}{2} + \varepsilon$, $w_i = \frac{1}{2} + \varepsilon$, $h_i = 1$, $i = 1, \ldots, m$,

m pieces of type T_b: $l_i = \frac{1}{2} - \varepsilon$, $w_i = \frac{1}{2} + \varepsilon$, $h_i = 1 - \varepsilon$, $i = m + 1, \ldots, 2m$,
m pieces of type T_c: $l_i = \frac{1}{2} + \varepsilon$, $w_i = \frac{1}{2} - \varepsilon$, $h_i = 1 - 2\varepsilon$, $i = 2\mu + 1, \ldots, 3m$,
m piece of type T_d: $l_i = \frac{1}{2} - \varepsilon$, $w_i = \frac{1}{2} - \varepsilon$, $h_i = 1 - 3\varepsilon$, $i = 3m + 1, \ldots, 4m$.

Obviously, $H_{\text{opt}} \leqq m$, since T_a, T_b, T_c and T_d can be packed in a layer of height 1.

The pieces $T_1, \ldots, T_{4m}$ are ordered with respect to decreasing height. According to the 3DBP algorithm we obtain:

a) $m - 1$	layers with exactly one piece T_a,	$H_k = 1$,
b) 1	layer with one T_a and one T_b,	$H_k = 1$,
c) $\dfrac{m-2}{2}$	layers with two pieces T_b,	$H_k = 1 - \varepsilon$,
d) 1	layer with one T_b and two T_c,	$H_k = 1 - \varepsilon$,
e) $\dfrac{m-4}{2}$	layers with two pieces T_c,	$H_k = 1 - 2\varepsilon$,
f) 1	layer with two T_c and one T_d,	$H_k = 1 - 2\varepsilon$,
g) $\dfrac{m-4}{4}$	layers with four pieces T_d,	$H_k = 1 - 3\varepsilon$,
h) 1	layer with three T_d,	$H_k = 1 - 3\varepsilon$.

Summing-up yields

$$\begin{aligned} H_{\text{NFD}} &= m + \frac{m}{2}(1 - \varepsilon) + \left(\frac{m}{2} - 1\right)(1 - 2\varepsilon) + \frac{m}{4}(1 - 3\varepsilon) \\ &= \frac{9}{4}m - 1 - \left(\frac{9}{4}m - 2\right)\varepsilon . \end{aligned}$$

Hence,

$$H_{\text{NFD}} \geqq \frac{9}{4} H_{\text{opt}} - 1 - \left(\frac{9}{4}m - 2\right)\varepsilon . \tag{12}$$

On the other hand, from (11) it follows:

$$H_{\text{NFD}} \leqq \frac{4}{1 - 4\varepsilon + 4\varepsilon^2} H_{\text{opt}} + 1 . \tag{13}$$

If ε tends to zero we have

$$\frac{9}{4} H_{\text{opt}} - 1 \leqq H_{\text{NFD}} \leqq 4H_{\text{opt}} + 1 .$$

Remark. The "large" gap between 9/4 and 4 corresponds to Fig. 1 and results from the "relative bad" estimation in (11) which is used for each layer. It is to denote that the lower bounds also depend on the range of the input data. Improvements of the quality of the performance bounds are only obtainable if improved estimations about the area utilization are known.

8. Conclusional remarks

The aim of this paper is to propose a first approximation algorithm for the three-dimensional bin packing problem. This algorithm is based on the NFD algorithm for two-dimensional bin packing. The 3DBP algorithm is asymptotically exact but further questions are still open. For instance, some estimations (bounds) are not as sharp as desired. Possibly, better results may be obtained if other packing strategies are used and/or sharper results about area utilization are found.

Acknowledgement

The author thanks the unknown referees for their valuable remarks and Prof. Dr. sc. nat. *J. Terno* (Dresden) for helpful discussions.

References

[1] *Berkey, J. O., P. Y. Wang*: Two-dimensional finite bin-packing algorithms. J. Oper. Soc. **38** (1985), 423–429.
[2] *Coffman, E. G., M. R. Garey, D. S. Johnson, R. E. Tarjan*: Performance bounds for level-oriented two-dimensional packing algorithms. SIAM J. Comput. **9** (1980), 808–826.
[3] *Coffman, E. G., G. S. Luecker, A. H. G. Rinnooy Kan*: Asymptotic methods in the probabilistic analysis of sequencing and packing heuristics. Management Sci. **34** (1988), 266–290.
[4] *Friesen, D. K., F. S. Kuhl*: Analysis of a hybrid algorithm for packing unequal bins. SIAM J. Comput. **17** (1988), 23–40.

Kurzfassung

In der Arbeit wird ein Näherungsalgorithmus für das dreidimensionale Bin-Packing-Problem vorgeschlagen und hinsichtlich der Güte der Näherungen im Vergleich zu optimalen Lösungen untersucht. Dazu wird ein modifizierter (zweidimensionaler) Bin-Packing-Algorithmus für beschränkte Bins bezüglich der erreichbaren Mindestflächenauslastung betrachtet. Abschließend wird anhand eines Beispiels eine untere Schranke für die Güte des Näherungsalgorithmus angegeben.

Резюме

В работе предлагается приближенный метод покрытия трехмерной полосы. Исследуется минимальное качество полученного приближенного решения по сравнению с оптимальным. С этой целью рассматривается модифицированный алгоритм покрытия двухмерной полосы с ограниченной шириной и оценивается его мера использования имеющейся площади снизу. Наконец, указывается нижняя граница для качества приближенного решения при помощи »неблагоприятного« примера.

(Received: first version May 30, 1990,
revised version November 14, 1990)

Author's address:

Dr. G. Scheithauer
Institute of Numerical Mathematics
Technological University Dresden
Mommsenstr. 13
O-8027 Dresden
Germany

J. Inform. Process. Cybernet. **EIK 27** (1991) 5/6, p. 272
(formerly: Elektron. Inform.verarb. Kybernet.)

Book Review

Anne Kaldewaij: Programming. The Derivation of Algorithms. (Prentice Hall International Series in Computer Science). Prentice-Hall, New York – London – Toronto – Sydney – Tokyo – Singapore 1990. 216 pp.; $ 31.95

This textbook on programming continues an approach originated by *C. A. R. Hoare's* article: "An axiomatic basis for computer programming" (Comm. ACM **10** (1969)) and refined by *E. W. Dijkstra* and *W. H. J. Feijen* during the 1970s and 1980s. Programming is considered as a stepwise activity of deriving algorithms from their specifications, thus, assuring the correctness of the final algorithms. Or in the words of the author: "A program together with its specification is viewed as a theorem". In each step, the programmer has to state creatively design decisions leading to a refinement of the program derived so far.

The idea is examplified by *Dijkstra's* guarded command language. A specification is expressed, by predicate calculus, as a precondition P and a postcondition Q of the program S: $\{P\}\ S\{Q\}$. The bottleneck problem of finding invariants (a kind of intermediate conditions) is discussed in greater detail.

The bigger part of this book is devoted to practicing this method by examples. Programming problems presented and handled are searching and sorting by different methods, manipulation and retrieval operations on arrays, and others.

This book is organized in a very didactic way: It is suitable for university teachers to prepare an introductory course to computer science as well as for students to use it in a private study of this subject. Many exercises are helpful.

K. Bothe