



(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
27.08.2003 Bulletin 2003/35

(51) Int Cl.7: **B65H 39/04**

(21) Application number: **03075501.1**

(22) Date of filing: **21.02.2003**

(84) Designated Contracting States:
**AT BE BG CH CY CZ DE DK EE ES FI FR GB GR
HU IE IT LI LU MC NL PT SE SI SK TR**
Designated Extension States:
AL LT LV MK RO

(72) Inventors:
• **Pothof-Kooi, Ellen Johanna**
2661 HA Bergschenhoek (NL)
• **Geeraerts, Johannes Stephanus Maria**
2631 WB Nootdorp (NL)

(30) Priority: **21.02.2002 NL 1020025**

(74) Representative: **Jorritsma, Ruurd et al**
Nederlandsch Octrooibureau
Scheveningseweg 82
P.O. Box 29720
2502 LS Den Haag (NL)

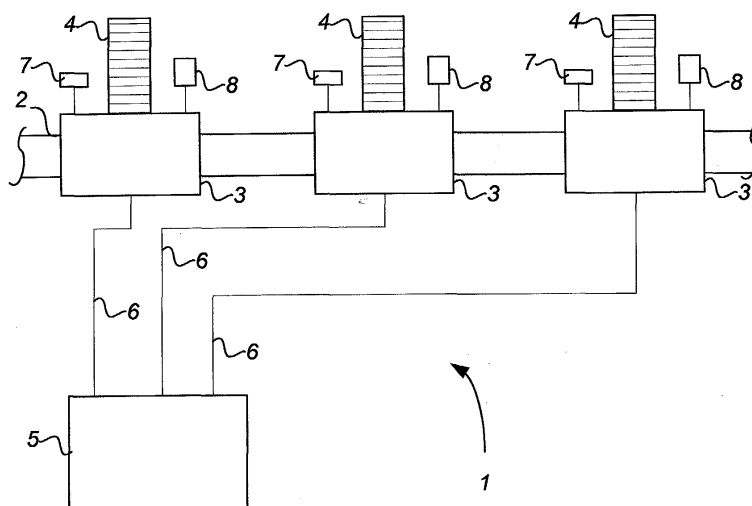
(71) Applicant: **Koninklijke PTT Post B.V.**
2500 GG Den haag (NL)

(54) **Installation and method for the selective production of stacks of printed matter**

(57) The present invention relates to an installation for collating stacks of printed products. The installation consists, inter alia, of a gathering machine, a predetermined number of feeders and a computer, the feeders being equipped to feed printed products to the gathering machine and stacks of printed products being produced in batches. Each batch consists of a set of groups of stacks where all stacks within a group have the same composition of printed products. The computer determines a feeder schedule in which it is specified which

feeder has to be activated, deactivated or changed over at what point in time. The computer of the invention also determines a (sub)optimum batch sequence. As a result of determining this (sub)optimum batch sequence in advance, the number of times that a feeder that has been used for a particular batch has to be changed over for a subsequent batch during the batch processing process will be as low as possible. As a result, the time for which the gathering machine has to be stopped is minimum.

FIG. 1



Description

[0001] The present invention relates to an installation for collating stacks of printed products comprising a gathering machine, a predetermined number of feeders and a computer, the feeders being equipped to feed printed products to the gathering machine and stacks of printed products being produced in batches, a batch consisting of a set of groups of stacks where all stacks within a group have the same composition of printed products, and the computer being equipped to determine a feeder schedule in which it is specified which feeder has to be activated, deactivated or changed over at what point in time.

[0002] An installation of the above mentioned type is disclosed, into alia, in International Patent Application WO 94/20400. The computer in the known installation is so equipped that the various feeders are controlled in such a way that the correct stacks of printed matter are produced. These stacks can be provided with a film at the end of the gathering machine and then transported to the final destination.

[0003] In the case of a machine having a limited number of feeders (that is to say a lower number of feeders than the number of different printed products in a production period) and an arbitrary processing sequence of successive batches, the content of the feeders must be regularly changed. Changing the content of the feeders, that is to say the removal of a stock of printed products and the replacement of this by a stock of new printed products, demands resetting of the feeder at virtually every change. This is time-consuming and disrupts the process.

[0004] The aim of the present invention is to provide an installation with which as few as possible new settings of the feeders are required during processing of the batches. To this end an installation as described in the preamble is provided, characterised in that the computer is equipped to carry out the following steps:

- a) receipt of data on the composition of the stacks of printed products;
- b) initialisation of a batch sequence;
- c) determination of a feeder schedule to achieve the batch sequence;
- d) storing a total number of change-overs in a variable (#CO_old);
- e) initialisation of a variable (MNC) to a positive integer, for dividing the batch sequence into part sequences, each part sequence comprising one or more batches and the variable (MNC) indicating the maximum number of change-overs of the feeders between successive batches in a part sequence;
- f) division of the batch sequence into the part sequences defined in this way, with the aid of the variable (MNC);
- g) rearrangement of the part sequences in the batch sequence;

h) reducing the value of the variable (MNC), and thus the size of the part sequences, if the total number of change-overs is not less than the value of the variable (#CO_old), or leaving the value of the variable (MNC) unchanged if the total number of change-overs is less than the value of the variable (#CO_old);

i) storage of the total number of change-overs in the variable (#CO_old);

j) repetition of the four preceding steps as long as the value of the variable (MNC) is greater than a specific minimum value (MNC_min);

k) provision of the feeders with the feeder schedule associated with the last batch sequence determined.

[0005] By determining an optimum (or sub-optimum) batch sequence in advance, the number of times that a feeder that has been used for a particular batch has to be changed over for a subsequent batch during the batch processing process will be as low as possible. As a result, the time for which the gathering machine has to be stopped is minimum. This yields a gain in time and the process is disrupted as little as possible.

[0006] The present invention also relates to a method for the collation of stacks of printed products with the aid of a gathering machine, a limited number of feeders and a computer, stacks of printed products being produced in batches, a batch consisting of a set of groups of stacks where all stacks within a group have the same composition of printed products, and a feeder schedule being determined in which it is specified which feeder has to be activated, deactivated or changed over at what point in time, characterised in that the method comprises the following steps;

- a) receipt of data on the composition of the stacks of printed products;
- b) initialisation of a batch sequence;
- c) determination of a feeder schedule to achieve the batch sequence;
- d) storing a total number of change-overs in a variable (#CO_old);
- e) initialisation of a variable (MNC) to a positive integer, for dividing the batch sequence into part sequences, each part sequence comprising one or more batches and the variable (MNC) indicating the maximum number of change-overs of the feeders between successive batches in a part sequence;
- f) division of the batch sequence into the part sequences defined in this way, with the aid of the variable (MNC);
- g) rearrangement of the part sequences in the batch sequence;
- h) reducing the value of the variable (MNC), and thus the size of the part sequences, if the total number of change-overs is not less than the value of the variable (#CO_old), or leaving the value of

the variable (MNC) unchanged if the total number of change-overs is less than the value of the variable (#CO_old);

i) storage of the total number of change-overs in the variable (#CO_old);

j) repetition of the four preceding steps as long as the value of the variable (MNC) is greater than a specific minimum value (MNC_min);

k) provision of the feeders with the feeder schedule associated with the last batch sequence determined.

[0007] In addition, the invention relates to a computer program that can be loaded by a computer that forms part of an installation for the collation of stacks of printed products, further comprising a gathering machine and a predetermined number of feeders, the feeders being equipped to feed printed products to the gathering machine and stacks of printed products being produced in batches, a batch consisting of a set of groups of stacks where all stacks within a group have the same composition of printed products, and the computer being equipped to determine a feeder schedule in which it is specified which feeder has to be activated, deactivated or changed over at what point in time, characterised in that the computer is equipped to carry out the above-mentioned steps a to k.

[0008] Finally, the invention relates to a data carrier provided with a computer program as described above.

[0009] Further advantages and characteristics of the present invention will become apparent on the basis of a description of an example, in which reference is made to the appended drawings, in which:

Fig 1 shows, diagrammatically, an installation for the collation of stacks of printed products;

Fig 2 shows a flowchart of the main procedure for the determination of the sequence of the batches;

Fig 3 shows a flowchart for the determination of the starting sequence of the batches;

Fig 4 shows a flowchart for the determination of the schedule for filling the feeders;

Fig 5 shows a flowchart for improving the batch sequence;

Fig 6A-6D show tables in which an example of a fill schedule for the feeders is shown;

Fig 7 shows, diagrammatically, a computer such as can be used in the invention.

[0010] In Figure 1 an installation 1 for the collation of stacks of printed matter such as can be used in the invention is shown. The installation 1 comprises, inter alia, a gathering machine 2, which, for example, is a conveyor belt. Furthermore, feeders 3 are present which feed printed products (hereinafter designated "folders") to the gathering machine 2. The installation 1 furthermore comprises a computer 5 that via communication means 6 is linked to displays 7 and indicators 8 that are installed

by the feeders 3. Although the communication means 6 are shown as continuous lines, these means can consist of wireless links just as well as of cables.

[0011] In one embodiment the batch processing process starts with the determination of the optimum sequence of the batches, hereinafter designated batch sequence. In the context of the invention a batch is defined as a set of groups of stacks where all stacks within a group have the same printed matter composition. In practice a group corresponds to a specific residential area in which a delivery person delivers a stack to every house.

[0012] The computer 5 is loaded by an operator with information on the folders that make up the various batches and the number of stacks in a batch. The computer 5 determines an optimum batch sequence on the basis of an algorithm. The algorithm is preferably determined by a program stored in a memory (see Figure 7). Feeder schedules belonging to this batch sequence are transmitted by the computer 5 to the various feeders 3. This information is preferably stored in PLCs (programmable logic controllers) present in the feeders 3. The feeders 3 are then filled with the folders 4 that make up the first batch. Filling is carried out by an installation operator on the basis of messages on the displays 7 and further directions on the indicators 8. Once the first batch has been processed, the computer 5 indicates on the displays 7 which folders have to be placed in the relevant feeders 3. The indicators 8 preferably comprise an indicator, such as a warning lamp, that alerts the user. This warning lamp can, for example, have the following functions:

- lamp on: the associated feeder is processing a specific folder. The folder that is being processed can be read off on the display 7 of the feeder.
- lamp flashing slowly: the folder in the feeder has to be changed. Which folder has to be loaded can be read off on the display 7.
- lamp flashing rapidly and machine stops: the user has not responded by means of a ready key on the feeder. The user must intervene.

[0013] After each (manual) change of folders, the feeder 3 must, if necessary, be mechanically adjusted to the new folders, after which a message that the feeder 3 is ready must be given. The ready message is preferably given by means of a push button, not shown, that is present on each feeder 3. If the operator presses the push button this means that the feeder 3 is ready for use.

[0014] The algorithm that is used by the computer 5 to determine the optimum batch sequence will be discussed on the basis of the following figures. In Figure 2 a flowchart for the main procedure is shown. The main procedure starts at step 20. This is followed by step 21, in which a starting sequence for the batches is determined. This step is a procedure that is elaborated in Figure 3. A specific batch sequence follows from step 21.

The next step is to determine a schedule for filling the feeders 3 on the basis of the batch sequence determined; step 22. This step 22 is elaborated in Figure 4. The number of change-overs of the feeders 3 that is needed to change from one batch to a subsequent batch follows from step 22. The batch sequence is now subdivided into blocks. Each block is a part sequence of the complete batch sequence. The size of a block (the number of batches) is determined by the "maximum number of change-overs" (MNC) variable. This variable is initialised in step 23 with an initial value MNC_init. A block ends where more change-overs are needed for the production of the following batch than the value that MNC indicates. A new block then starts. The start of the complete batch sequence is the start of the first block. The end of the complete batch sequence is the end of the last block.

[0015] After the initialisation of MNC, the batch sequence is divided into blocks in step 24. Because the block size is determined by the maximum number of change-overs of the feeders 3 from one batch to the subsequent batch, the blocks will differ in size.

[0016] The next step consists of a procedure for improving the batch sequence; see step 25. This step is elaborated in Figure 5. In step 26 an assessment is made as to whether a better sequence has been found. A better sequence is that in which the total number of change-overs is less than that in the previous sequence. If a better sequence has been found, the procedure returns to step 24. If this is not the case, step 27 then follows. In step 27 a test is carried out to determine whether the variable MNC is greater than a minimum MNC (MNC_min), where MNC_min is an integer having, for example, the value 1 or 2. If this is the case, step 28 then follows in which MNC is reduced by 1. Following step 28, step 24 is repeated. If MNC is equal to MNC_min in step 27, the main procedure is terminated; step 29. The batch sequence that has been determined last is used for the batch processing process.

[0017] The procedure 21 for determining the starting sequence is shown in Figure 3. After the start in step 30, the step for selecting the batch containing the most folders first follows; see step 31. If there are several batches containing the largest number of folders, the first batch of these that is found is then selected. This batch becomes the first batch in the sequence. The content of the feeders 3 is now determined. This is easy because all feeders 3 are empty at the start. In step 32 it is then investigated whether there are batches that can be run with the content of the feeders 3 determined in this way. These batches are placed in the sequence after the first batch. If it is found in step 33 that all batches can be placed in this way, the procedure is complete; see 38. If it is not possible to place all batches, step 34 follows. In step 34, a parameter v is initialised with the value one. A search is then made in the batches that have not yet been placed to find a batch that can be run with v change-overs of the feeders 3; see step 35. If this is not

successful, the parameter v is then increased by one in step 36. Step 35 is then repeated. If there is a batch that can be run with v change-overs of the feeders 3, step 37 then follows in which the content of the feeders 3 is determined in such a way that the new batch determined in step 35 can be run. In this context placing folders in empty feeders 3 is *not* regarded as relevant change-overs. Placing a folder in a filled feeder 3 that is not used in a preceding batch is referred to as a so-called "soft" change-over. In contrast to the so-called "hard" change-overs, the machine does not have to be stopped for a "soft" change-over if the relevant feeder 3 is changed over in good time. However, it does generate work for the user of the machine. Thus, the aim must be for as few as possible "hard" and "soft" change-overs.

[0018] If folders have to be changed in step 37 in order to run the new batch, the choice of the feeders 3 to be changed is then arbitrary, provided the folders that are needed for the batch to be added are already in a specific feeder 3. This (these) feeder(s) 3 is (are) then not changed. Step 32 is then carried out again, after which step 33 follows. If it is found in step 33 that all batches have been placed, the procedure is complete; see 38. If, on the other hand, it is found in step 33 that all batches have not yet been completed, the program then continues with steps 34, 35, 36, 37, 32 and 33 until all batches have been completed. The computer 5 has then determined a starting sequence and stored this in its memory.

[0019] The procedure 22 for determining the schedule for filling the feeders 3 is shown in Figure 4. After the start in step 40 the folders for the first batch from the starting sequence as determined with the aid of the schedule in Figure 3 are (virtually) placed in the feeders 3 in the first step; see step 41. A check is then made to determine whether there is a subsequent batch in the sequence; see step 42. If there is a subsequent batch, step 43 is then carried out. In step 43 the first folder of the batch is selected. This folder is termed the "current folder" in the chart. In step 44 it is investigated whether this current folder is already (virtually) present in a feeder. If this is not the case, the program proceeds to step 45. In step 45 a check is carried out to determine whether a feeder is free. If this is the case, step 46 then follows in which the current folder is placed in an empty feeder. For the sake of clarity it is pointed out here that the wording "filling a feeder with a folder" is used to denote that the feeder 3 is filled with a predetermined quantity of identical folders.

[0020] Step 47, in which a check is carried out to determine whether the current folder is the last folder from the batch, follows step 46. If this is not the case, step 48 follows. In this step the next folder in the batch is selected, indicated mathematically by "current folder = next folder in batch", after which the program proceeds to step 44. If it is found in step 47 that the current folder is the last folder in the batch, the program returns to step 42.

[0021] If it is found in step 45 that no feeder is free, it

is then determined in step 49 which folder that is currently present in the feeders 3 is the last needed again for a later batch. This method is analogous to the known "keep tool needed soonest" algorithm. The folder determined in this way is removed from the feeder 3 in step 50. The current folder is then placed in this empty feeder 3 in step 46. If it is found in step 42 that all batches have been dealt with, the procedure is terminated, see 51, and the computer 5 has determined a schedule for filling the feeders 3. The computer 5 stores this schedule.

[0022] Figure 5 shows a flowchart for a procedure 25 for improving the sequence. This is necessary because the batch sequence that is associated with the schedule for filling the feeders 3, as determined on the basis of the flowchart in Figure 4, is not necessarily optimum. The optimum batch sequence is that in which as few change-overs of the feeders 3 as possible are needed. After the start in step 60, the first block of the sequence that followed from Figure 4 is selected in step 61. Furthermore, the first possible location in the sequence in which a block can be inserted is now determined. Insertion locations are between the various blocks and right at the start and at the end of the sequence. The first possible insertion location for the first block is between the second and the third block. The first possible insertion location for the second block is right at the front of the sequence. In one embodiment of the invention, restrictions with regard to the time of production are taken into account when determining the possible insertion location; certain batches have, for example, to be produced before others.

[0023] In step 62 the block from step 61 is moved to the insertion location from step 61. In step 63 the schedule for the feeders 3 is then determined with the aid of the procedure from Figure 4. In step 64 it is determined whether the new sequence has a lower total number of change-overs. If this is the case this is regarded as a better sequence and the next block is selected in step 67 (provided there is still a block; this is checked in step 66). After step 67, step 62 is carried out again with the new block and the first possible insertion location. If it follows from step 64 that the new sequence is not an improvement, the next insertion location is then determined in step 68 (provided there is still an insertion location; this is checked in step 65). After step 68, step 62 is then carried out again with the same block and a new insertion location. If it is found in step 66 that the final block has been reached, the procedure is terminated in step 69.

[0024] On the basis of an example as shown in Figures 6A, 6B, 6C and 6D it will now be explained how the procedures described above function. In Figure 6A a table has been drawn in which the columns correspond to five feeders called F1, ..., F5. The number of hard change-overs, the number of soft change-overs and the total number of change-overs, respectively, are given in the last three columns. The rows in the table correspond to a number of batches, i, e, B1, ..., B12. A batch Bx

consists of a sub-set of the total number of products (read folders) P1, ..., PN. Here N=12, but N can be any arbitrary other integer. In this example a batch does not contain more than four products. Before determining the starting sequence, see step 21 in Figure 2, it is first determined which batch contains the most folders; see step 31 in Figure 3. This batch is called B1 with, as content, the products P1, P2, P3, P4. The batches that can be run with the current content of the feeders 3 are now added in step 32. This is batch B2 with P2, P3 as content.

[0025] The parameter v is then initialised with the value one in step 34. A batch that can be run with v=1 change-overs of the feeders 3 is now sought. This is batch B3 with P3, P4, P5 as content. The content of the feeders 3 is then determined in step 37 so that batch B3 can be run. It can be seen Figure 6A that the products for batch B3 can be (virtually) placed in the feeders F3, F4, F5. The first two products of batch B3 were already in the feeders 3. Product P5 is placed in an empty feeder F5. The product P1 remains in feeder F1 but feeder F1 is switched to the stop position. The same applies in respect of feeder F2, which has product P2 as content. Because P5 is placed in an empty feeder, there is no relevant change-over in this case.

[0026] In step 32 the batches that can be run with the new content of the feeders 3 are then sought. No batch meets these criteria. It is now investigated in step 35 whether there is a batch that can be run with v=1 change-overs of the feeders 3. Batch B4 meets these criteria and has P3, P5, P6 as content. The products P3, P5 are already in the machine in feeders F3 and F5, respectively, but the other product P6 is placed in feeder F1, which thus has to be changed-over for this purpose. Since product P6 can be placed in the feeder F1 in advance, there is a soft change-over here.

[0027] Batch B5 can be run with the current feeder content and is thus the next batch in the sequence. There are no further batches that can be run with the current feeder content, so that, once again, the value for v is initialised at 1 in step 34. We now find batch B6 which can be run by placing product P7 in the feeders 3.

[0028] With the aid of step 32, batch B7 is found because this batch can be run without change-overs on the machine. It is now investigated in step 35 whether there is a batch that can be run with v=1 change-overs of the feeders 3. In this example there is no longer a batch that meets these requirements. In step 36, v is now increased by 1. Even with v=2 no batch can be found. Only with v=3 is batch B8 found with P5, P8, P9, P10 as content. The product P5 is already present on the machine in feeder F5, but the other three products are placed in feeders F1, F2, F3, which have to be changed-over for this purpose. Because feeder F1 and feeder F2 are in use during batch B7, there is a hard change-over in these cases. Feeder F3 is not in use, so the product P10 can be placed in feeder F3 in advance, as a result of which this is a soft change-over. The hard

change-overs are indicated by an "*" and the total number of change-overs is shown in the column with the heading "Total". It would also have been possible to place the products P8, P9, P10 in feeder F4. At this point in time it is not yet known what the best choice is. Note: in Figure 6A, the products that can optionally remain in the feeder(s) 3 after running a batch are not shown.

[0029] Now that the content of the feeders 3 is known again it is possible to investigate in step 32 which batches can be run with this content of the feeders. No batch meets these requirements; therefore, in step 35 batches are sought that can be run with $v=1$. Batch B9 can be run with one change-over, so that the product P4 is removed from feeder F4 in favour of product P11. It would also have been possible to choose to remove product P5, but at this point in time it is not yet known what is the best choice.

[0030] No batch that can be run without change-overs is found with the aid of step 32, so once again the value for v is initialised at 1 in step 34. No batches that can be run are found in step 35, so in step 36 the value for v is increased to 2. Batch B10 is now found, which can be run after carrying out two change-overs (a hard and a soft change-over). It is then concluded in step 33, just as in previous steps, that all batches have not been completed, so once again $v=1$ is initialised in step 34, so that the batch B11 is found in step 35. This batch is added to the sequence in step 32. In step 33 it is now found that batch B12 still has to be placed. This is added in the steps 33, 34, 35, 37, 32, the number of change-overs v being 1; see Figure 6A. On returning to step 33 it is found that all batches have now been completed and the end of the procedure has been reached; see step 38. This means that the starting sequence has now been determined; see step 21 in Figure 2. In Figure 6A the number of change-overs that have to take place in order to carry out this batch sequence is shown at the bottom of the "Total" column. The total number of change-overs in this example is 10.

[0031] The schedule for the feeders 3 is then determined in step 22 on the basis of the procedure shown in Figure 4. This gives the feeder schedule as shown in Figure 6B. Only the functioning of step 49 is explained here. In Figure 6B it can be seen that the content of the feeders 3 has been changed for running batch B4. In order to be able to run batch B4, the product P6 must be placed in a feeder. During batch B3 the product P1 is in feeder F1 and product P4 in feeder F4. Product P1 is used in batch B10 and product P4 in batch B5, whilst product P2 is used in batch B12. Product P2 is thus needed again later than product P1 and P4 and therefore product P2 is removed from feeder F2. The required product P6 is now placed in feeder F2. Batch B5 can be produced because all products for batch B5 are already in the feeders 3. At the start of batch B6 product P7 has to be placed in the feeders 3. Since the product P3 is the last needed (i.e. in batch B11), this is removed from feeder F3 and replaced by product P7 (hard change-

over).

[0032] Before the start of batch B8 a number of products have to be removed once again. Product P5 is already in feeder F5 and this is needed for batch B8 and thus remains in feeder F5. Products P1, P4, P6 and P7 can all be considered for removal. Products P4, P6 and P7 are no longer needed for the remaining batches and can thus be removed. The products P8, P9 and P10 are now placed in the feeders F2, F3 and F4, respectively, as a result of which three hard change-overs take place. For batch B9 the product P11 must be placed in a feeder 3. Feeder F1 and feeder F5 are both candidates for change-over. Since product P1, which is in feeder F1, is still needed later, product P5 in feeder F5 must make way for the product P11. For batch B10 the product P12 must be placed in a feeder. Both product P8 and product P10 are no longer needed for later batches and it is therefore chosen to remove product P8. Batch 11 consists, inter alia, of the product P3, which is not in the feeders 3. A choice is made to remove product P10 because this gives a soft change-over. The product P2 has to be placed for batch B12. A choice has been made to remove product P9 from feeder F3 in order to achieve this, whilst a choice could also have been made to remove product P1 or P3, but this would have resulted in hard change-overs. In Figure 6B it can now be seen that the total number of change-overs is nine.

[0033] The next step that is carried out is step 23 in Figure 2. In this example MNC_init is tw and MNC_min is zero. In step 24 the batch sequence is divided into blocks. The first block S 1 consists of the batches B to B7. After all, there are three change-overs between B7 and B8 and this is more than $MNC_init=2$. The second block S2 consists of batch B7 to B12 because there is always at most only one change-over between these batches.

[0034] Step 25, in which the batch sequence is improved, follows after step 24. This improvement procedure is shown in Figure 5. In step 61 the first block is selected. This is block S1. A first insertion location is also chosen. This is the location after S2 (in this example there is no other possibility). In step 62 block S1 is now removed and inserted in the insertion location concerned. This gives the batch sequence as is shown in Figure 6C. Here again "*" indicates where the hard change-overs take place. The total number of change-overs now amounts to ten.

[0035] The schedule for the feeders is now determined in step 63 in Figure 5. The procedure is not discussed again here. The end result of step 63 can be seen in Figure 6D. The number of change-overs is eight. This is a better sequence than that shown in Figure 6B. (Note: the number of change-overs is also shown in Figures 6A and 6C, but is actually superfluous and not of importance for the decision in step 64.) Because the response to the question in step 64 is in the affirmative and all possible block sequences have been investigated for improvements, the scheme in Figure 5 is exited

via step 66 and step 69. Step 26 in Figure 2 now follows. In step 26 the program now returns to step 24. The batch sequence is now divided into blocks again. With $MNC=2$ only one block is now produced. This cannot be further optimised. Thus, the answer in step 26 is "no". Therefore the program now proceeds to step 28 where MNC is reduced to one; see Figure 2. The division into blocks follows in step 24. This gives the blocks $S1'$ and $S2'$, where $S1'$ consists of the batches $B8$ and $B9$, whilst $S2'$ is determined by the batches $B10$ to $B7$; see Figure 6D. The procedure now continues with steps 25 and 26. After a number of iterations no further improvement will be found, so that the procedure will be terminated; see step 29.

[0036] It will be clear that in the case of a large batch sequence and many different products the method described above cannot always be guaranteed to yield an optimum solution. Very frequently a sub-optimal solution will be found. This is characteristic of a heuristic approach. The advantage of this heuristic approach is, however, that the computation time of the computer 5 remains restricted. In order to limit further restriction of this computation time the value of MNC_min can be varied. This can, for example, be greater than zero so that the procedure described above possibly finds a less good solution but where the computing time will be appreciably restricted.

[0037] Figure 7 shows a computer installation 5 with a processor 101 for carrying out mathematical processing such as is used in one embodiment. The processor 101 is connected to a number of memory components including a hard disc 105, read-only memory (ROM) 107, electrically erasable programmable read-only memory (EEPROM) 109 and random access memory (RAM) 111. Not all of these types of memory necessarily have to be present. Moreover, they do not have to be physically located close to the processor 101. They can also be located remotely.

[0038] The processor 101 is also connected to means for the input of instructions, data, etc. by a user, such as a keyboard 113 and a mouse 115. Other input means, such as a touch screen, a trackball and/or speech converter, which are known to those skilled in the art, can also be used.

[0039] A reader 117 connected to the processor 101 is provided. The reader 117 is equipped to read data from, and optionally to store data on, a data carrier, such as a floppy disc 119 or a CD-ROM 121. Other data carriers can be, for example, DVDs as is known to those skilled in the art.

[0040] The processor 101 is also connected to a printer 123 for printing output data on paper, as well as to a display unit 103, for example a monitor or LCD (liquid crystal display) screen, or any other type of display unit known to those skilled in the art. The processor 101 is equipped to communicate with the PLCs, optionally with the displays 7 or with the indicators 8 that are installed by the feeders 3, by means of input/output means 125

and via the communication means 6.

[0041] The processor 101 can have been implemented as a stand-alone system or as a number of processors operating in parallel, each of which is equipped to carry out sub-tasks of a larger program, or as one or more main processors with various sub-processors.

[0042] In a further embodiment the optimisation process is carried out by a computer that is not connected to the feeders 3. The result of the optimisation process is then stored on, for example, a data carrier and the data are then read in by the computer 5 that communicates with the feeders 3.

[0043] It will be clear that the invention can also be used when sorting other printed products, such as, for example, newspapers, instead of folders. The invention can, moreover, be applied to the sorting of products of a more general nature. Consideration can be given to the (optionally free) production and distribution of stacks of CDs. Yet another possibility is the distribution of products that are not stacked but are collected together, for example in a plastic bag, and then distributed.

Claims

1. Installation for collating stacks of printed products comprising a gathering machine (2), a predetermined number of feeders (3) and a computer (5), the feeders (3) being equipped to feed printed products (4) to the gathering machine (2) and stacks of printed products being produced in batches, a batch consisting of a set of groups of stacks where all stacks within a group have the same composition of printed products, and the computer being equipped to determine a feeder schedule in which it is specified which feeder (3) has to be activated, deactivated or changed over at what point in time, **characterised in that** the computer is equipped to carry out the following steps:

- a) receipt of data on the composition of the stacks of printed products;
- b) initialisation of a batch sequence;
- c) determination of a feeder schedule to achieve the batch sequence;
- d) storing a total number of change-overs in a variable ($\#CO_old$);
- e) initialisation of a variable (MNC) to a positive integer, for dividing the batch sequence into part sequences, each part sequence comprising one or more batches and the variable (MNC) indicating the maximum number of change-overs of the feeders (3) between successive batches in a part sequence;
- f) division of the batch sequence into the part sequences defined in this way, with the aid of the variable (MNC);
- g) rearrangement of the part sequences in the

batch sequence;

h) reducing the value of the variable (MNC), and thus the size of the part sequences, if the total number of change-overs is not less than the value of the variable (#CO_old), or leaving the value of the variable (MNC) unchanged if the total number of change-overs is less than the value of the variable (#CO_old);

i) storage of the total number of change-overs in the variable (#CO_old);

j) repetition of the four preceding steps as long as the value of the variable (MNC) is greater than a specific minimum value (MNC_min);

k) provision of the feeders (3) with the feeder schedule associated with the last batch sequence determined.

2. Installation according to claim 1, **characterised in that** the step for the initialisation of the batch sequence comprises the following steps:

a) determining a largest batch in which the stacks contain the most printed products and adding this as first batch to the batch sequence;

b) adding to the batch sequence all batches that can be run after the preceding step without change-overs being required;

c) adding to the batch sequence a specific batch that can be run after the preceding step with as few change-overs as possible;

d) repeating the previous two steps until all batches have been added to the batch sequence.

3. Installation according to one of the preceding claims, **characterised in that** the step for determining a feeder schedule to achieve the batch sequence comprises the following steps:

a) assignment of the printed products of the first batch of the batch sequence to the feeders (3);

b) determination of a first printed product of a subsequent batch in the batch sequence and definition of this as current printed product;

c) assignment of the current printed product to a feeder (3) in which the current printed product is already present, or to a feeder (3) that is still empty, or to a feeder (3) in which another printed product is present, wherein, in the latter case, that feeder (3) is chosen in which a printed product is present that, compared with the printed products that are in the other feeders (3), has to be placed last in a feeder (3) again in subsequent batches;

d) repetition of the preceding step with a subsequent printed product as current printed product, until all printed products of the next batch have been placed;

e) repetition of the three preceding steps until all batches of the batch sequence have been dealt with.

4. Installation according to one of the preceding claims, **characterised in that** the step for rearrangement of the part sequences in the batch sequence comprises the following steps:

a) selection of a first part sequence;

b) determination of a first possible insertion location;

c) moving the selected part sequence to the insertion location determined;

d) determination of the feeder schedule;

e) calculation of the total number of change-overs in the batch sequence;

f) selection of a subsequent part sequence if the total number of change-overs has reduced, or determination of a next insertion location if the total number of change-overs has not reduced;

g) repetition of the four preceding steps until all part sequences have been dealt with.

5. Installation according to Claim 4, **characterised in that** the next possible insertion location is a location between two successive part sequences, right at the front of the batch sequence or right at the back of the batch sequence.

6. Method for the collation of stacks of printed products with the aid of a gathering machine (2), a limited number of feeders (3) and a computer (5), wherein stacks of printed products are produced in batches, a batch consisting of a set of groups of stacks where all stacks within a group have the same composition of printed products, and a feeder schedule is determined in which it is specified which feeder (3) has to be activated, deactivated or changed over at what point in time, **characterised in that** the method comprises the following steps;

a) receipt of data on the composition of the stacks of printed products;

b) initialisation of a batch sequence;

c) determination of a feeder schedule to achieve the batch sequence;

d) storing a total number of change-overs in a variable (#CO_old);

e) initialisation of a variable (MNC) to a positive integer, for dividing the batch sequence into part sequences, each part sequence comprising one or more batches and the variable (MNC) indicating the maximum number of change-overs of the feeders (3) between successive batches in a part sequence;

f) division of the batch sequence into the part

sequences defined in this way, with the aid of the variable (MNC);

g) rearrangement of the part sequences in the batch sequence;

h) reducing the value of the variable (MNC), and thus the size of the part sequences, if the total number of change-overs is not less than the value of the variable (#CO_old), or leaving the value of the variable (MNC) unchanged if the total number of change-overs is less than the value of the variable (#CO_old);

i) storage of the total number of change-overs in the variable (#CO_old);

j) repetition of the four preceding steps as long as the value of the variable (MNC) is greater than a specific minimum value (MNC_min);

k) provision of the feeders (3) with the feeder schedule associated with the last batch sequence determined.

5

10

15

20

7. Computer program that can be loaded by a computer (5) that forms part of an installation for the collation of stacks of printed products, further comprising a gathering machine (2) and a predetermined number of feeders (3), the feeders being equipped to feed printed products (4) to the gathering machine (2) and stacks of printed products being produced in batches, a batch consisting of a set of groups of stacks where all stacks within a group have the same composition of printed products, and the computer being equipped to determine a feeder schedule in which it is specified which feeder (3) has to be activated, deactivated or changed over at what point in time, **characterised in that** the computer (5) is equipped to carry out the following steps:

25

30

35

a) receipt of data on the composition of the stacks of printed products;

b) initialisation of a batch sequence;

c) determination of a feeder schedule to achieve the batch sequence;

d) storing a total number of change-overs in a variable (#CO_old);

e) initialisation of a variable (MNC) to a positive integer, for dividing the batch sequence into part sequences, each part sequence comprising one or more batches and the variable (MNC) indicating the maximum number of change-overs of the feeders (3) between successive batches in a part sequence;

40

45

50

f) division of the batch sequence into the part sequences defined in this way, with the aid of the variable (MNC);

g) rearrangement of the part sequences in the batch sequence;

55

h) reducing the value of the variable (MNC), and thus the size of the part sequences, if the total number of change-overs is not less than

the value of the variable (#CO_old), or leaving the value of the variable (MNC) unchanged if the total number of change-overs is less than the value of the variable (#CO_old);

i) storage of the total number of change-overs in the variable (#CO_old);

j) repetition of the four preceding steps as long as the value of the variable (MNC) is greater than a specific minimum value (MNC_min);

k) provision of the feeders (3) with the feeder schedule associated with the last batch sequence determined.

8. Data carrier provided with a computer program according to Claim 7.

FIG. 1

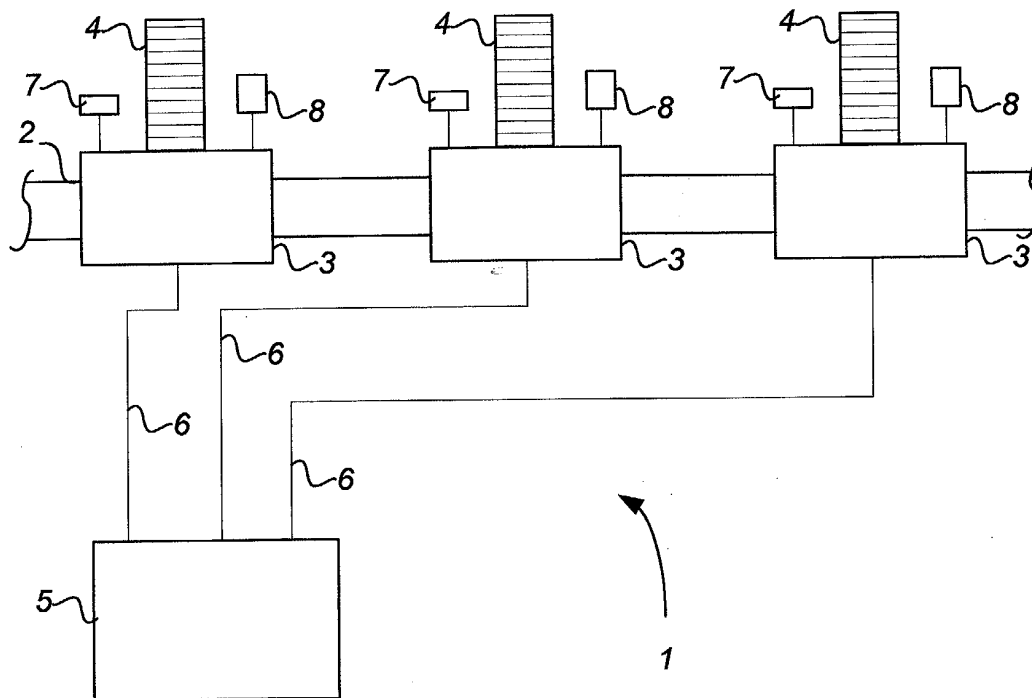
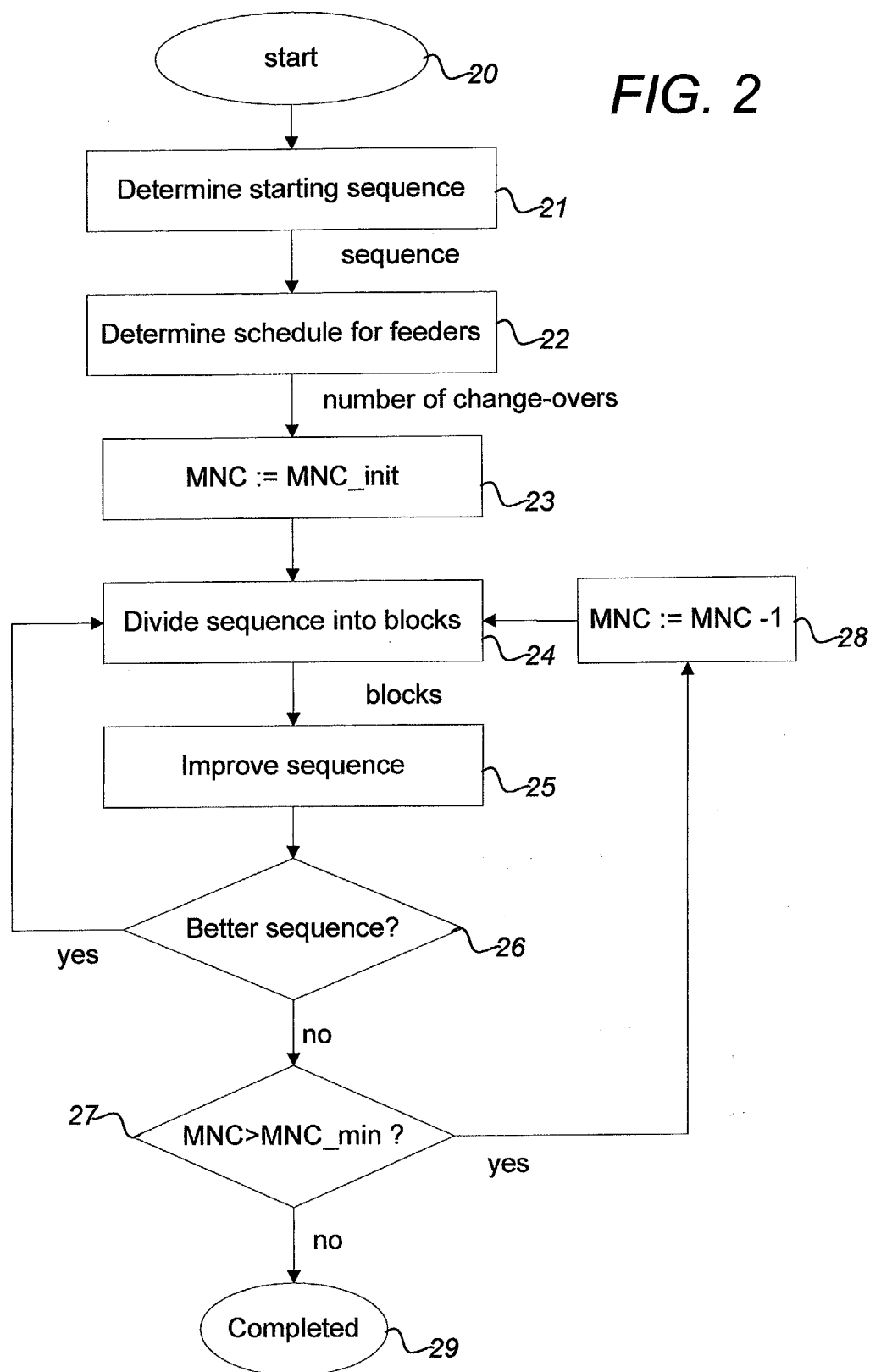


FIG. 2



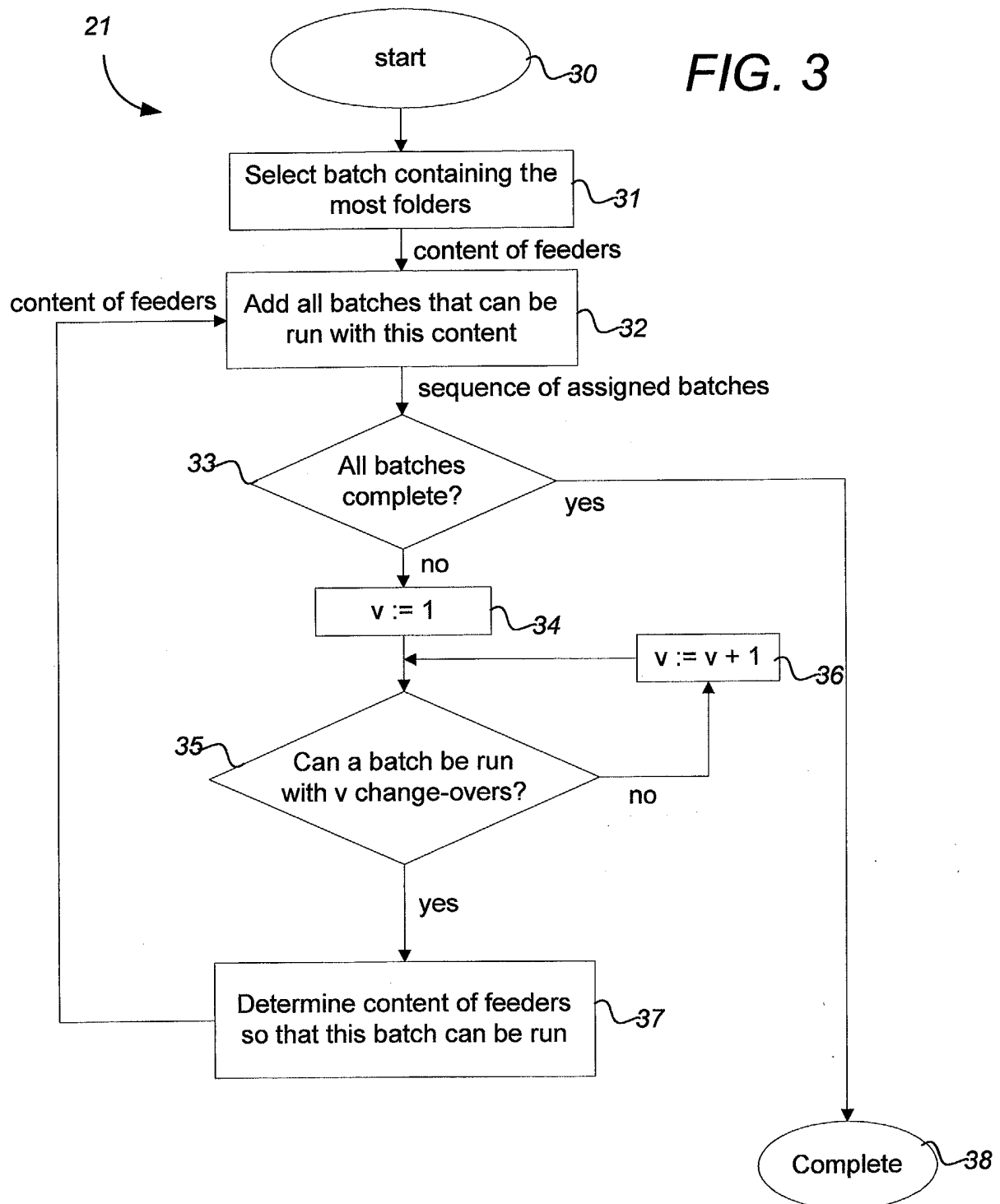


FIG. 4

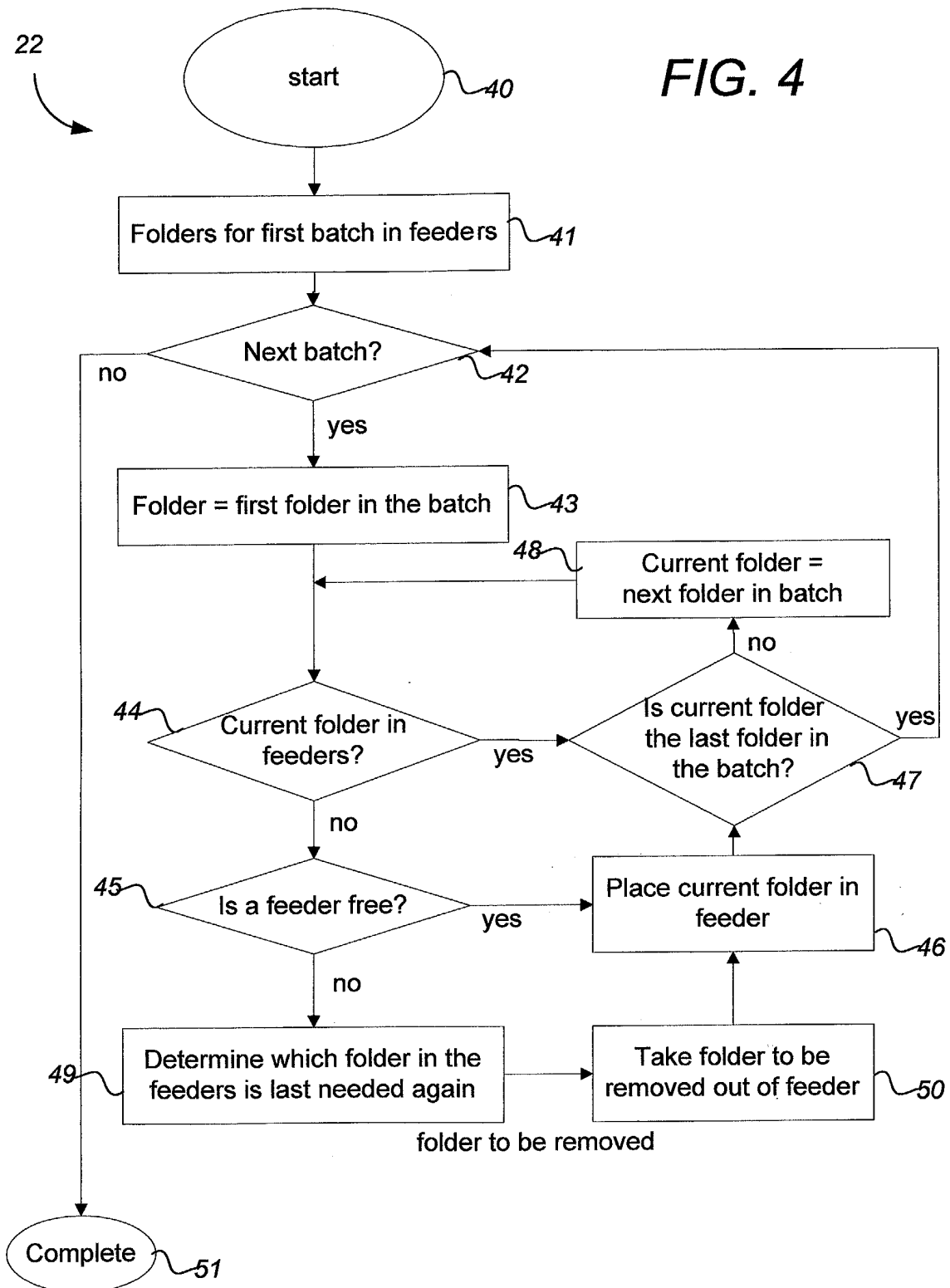


FIG. 5

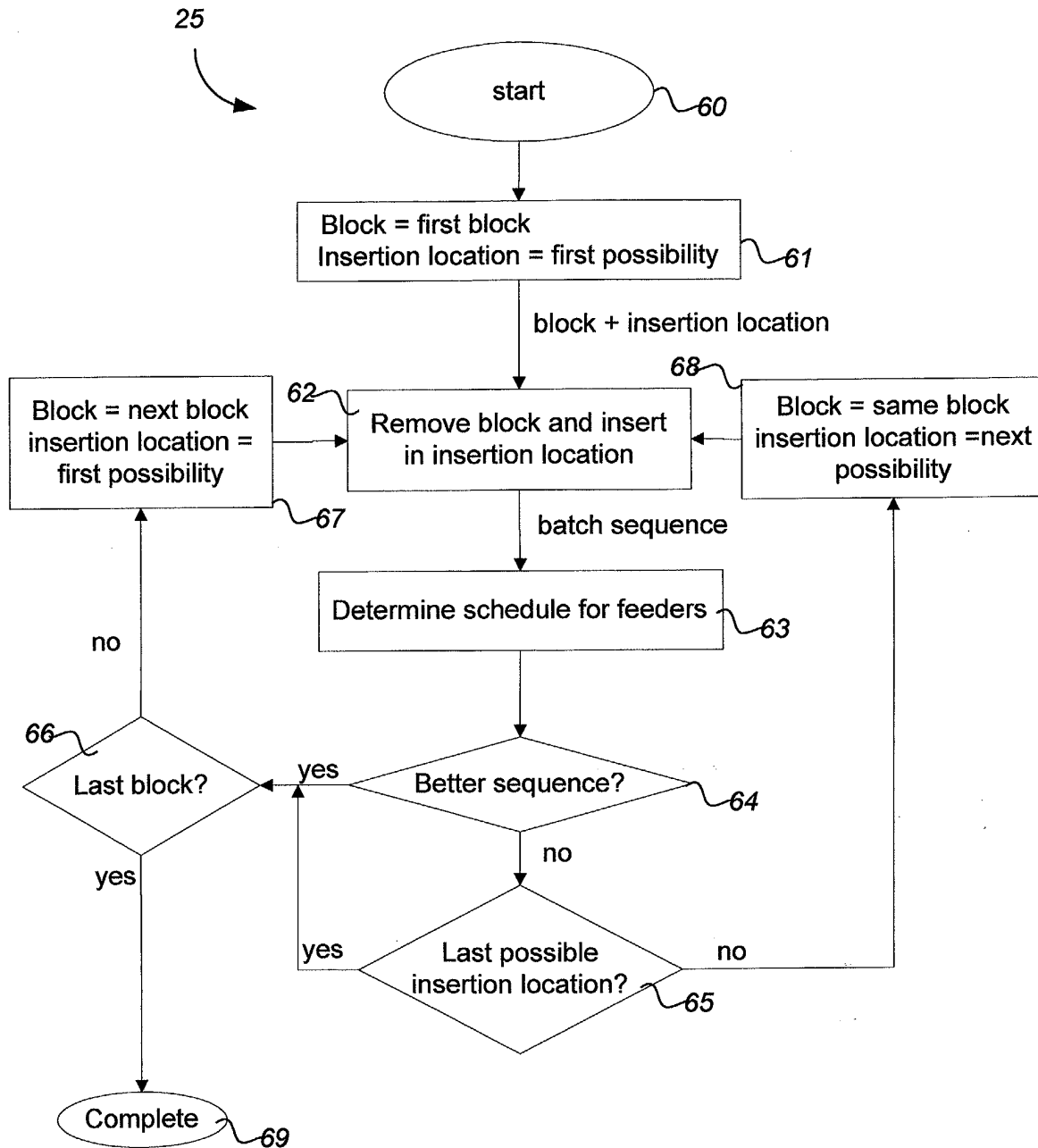


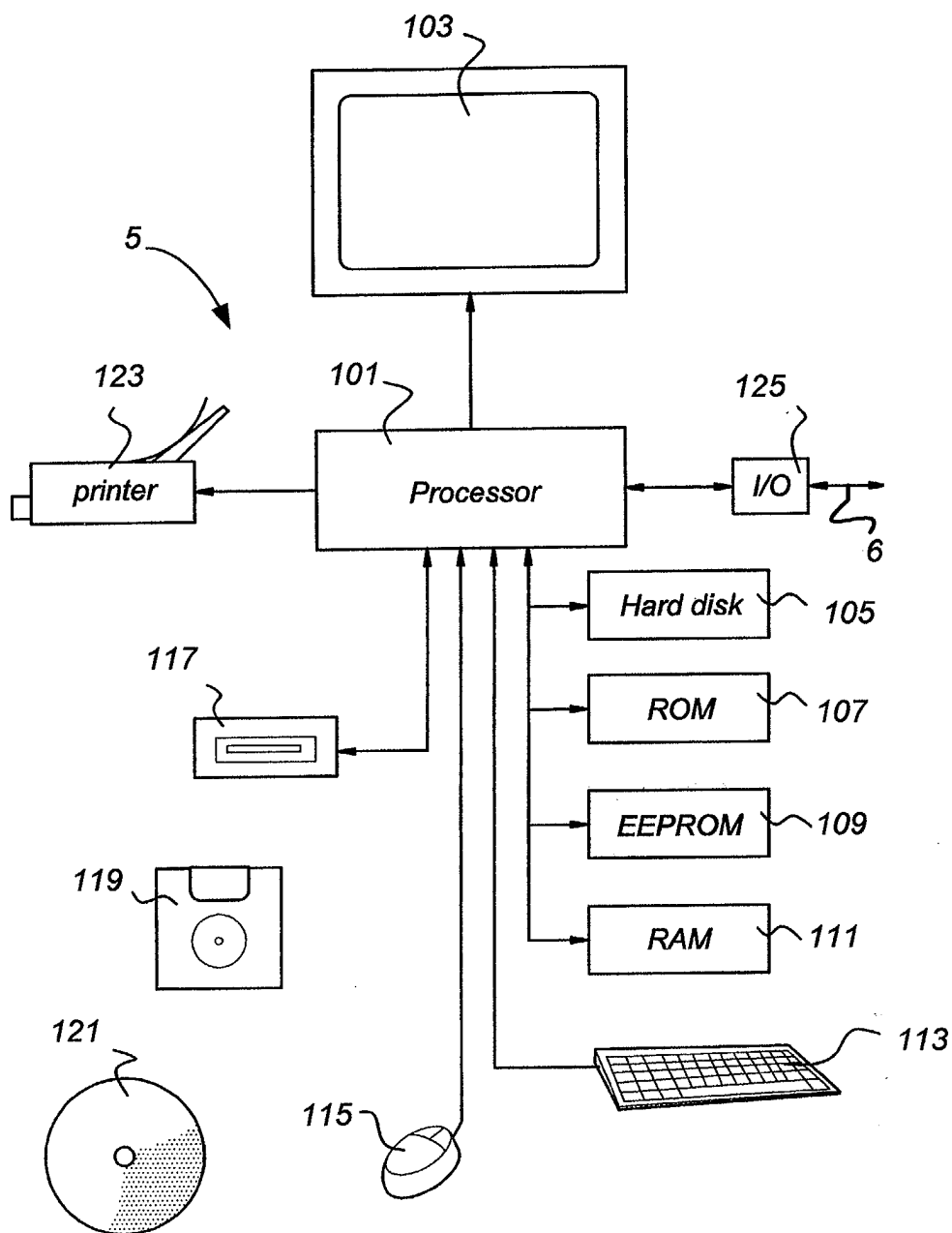
Fig. 6A

BATCH		FEEDER					CHANGE-OVERS		
		F1	F2	F3	F4	F5	# Hard	# Soft	Total
B1		P1	P2	P3	P4				
	B2		P2	P3			0	0	0
B3				P3	P4	P5	0	0	0
B4		P6		P3		P5	0	1	1
	B5	P6		P3	P4		0	0	0
B6		P6	P7		P4		0	1	1
	B7	P6	P7		P4		0	0	0
B8		P8*	P9*	P10		P5	2	1	3
B9		P8	P9	P10	P11		0	1	1
B10		P1*	P9			P12	1	1	2
B11		P1	P3*		P11	P12	1	0	1
B12				P2	P11	P12	0	1	1
							4	6	10

Fig. 6B

		FEEDER					CHANGE-OVERS		
BATCH		F1	F2	F3	F4	F5	# Hard	# Soft	Total
B1		P1	P2	P3	P4				
	B2		P2	P3			0	0	0
B3				P3	P4	P5	0	0	0
B4			P6	P3		P5	0	1	1
	B5		P6	P3	P4		0	0	0
B6			P6	P7*	P4		1	0	1
	B7		P6	P7	P4		0	0	0
B8			P8*	P9*	P10*	P5	3	0	3
B9			P8	P9	P10	P11*	1	0	1
B10		P1	P12*	P9			1	0	1
B11		P1	P12		P3	P11	0	1	1
B12			P12	P2		P11	0	1	1
							6	3	9

FIG. 7





European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 03 07 5501

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
D,A	WO 94 20400 A (GRAPHIA HOLDING AG ; LEHMANN PETER (CH); LUETHI ERNST (CH)) 15 September 1994 (1994-09-15) * the whole document *	1,6	B65H39/04
A	WO 98 04364 A (OPEX CORP) 5 February 1998 (1998-02-05) * the whole document *	1,6	
A	EP 0 804 974 A (PITNEY BOWES) 5 November 1997 (1997-11-05) * the whole document *	1,6	
A	US 5 346 196 A (ARMSTRONG MICHAEL J ET AL) 13 September 1994 (1994-09-13) * the whole document *	1,6	
A	US 4 253 651 A (MCINERNY GEORGE P ET AL) 3 March 1981 (1981-03-03) * column 8, line 35 - column 9, line 10 * * figure 11 *	1,6	
The present search report has been drawn up for all claims			TECHNICAL FIELDS SEARCHED (Int.Cl.7) B65H B07C B42C
Place of search THE HAGUE		Date of completion of the search 19 June 2003	Examiner Pussemier, B
CATEGORY OF CITED DOCUMENTS X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document			

EPO FORM 1503 03.02 (P04C01)

**ANNEX TO THE EUROPEAN SEARCH REPORT
ON EUROPEAN PATENT APPLICATION NO.**

EP 03 07 5501

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

19-06-2003

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9420400 A	15-09-1994	WO 9420400 A1	15-09-1994
		EP 0639154 A1	22-02-1995
WO 9804364 A	05-02-1998	US 5842577 A	01-12-1998
		AU 3736297 A	20-02-1998
		DE 69718471 D1	20-02-2003
		EP 0912258 A1	06-05-1999
		US 2002104782 A1	08-08-2002
		WO 9804364 A1	05-02-1998
		US 6112902 A	05-09-2000
		US 6311846 B1	06-11-2001
EP 0804974 A	05-11-1997	US 5777883 A	07-07-1998
		DE 69706356 D1	04-10-2001
		DE 69706356 T2	29-05-2002
		EP 0804974 A1	05-11-1997
US 5346196 A	13-09-1994	WO 9420401 A1	15-09-1994
US 4253651 A	03-03-1981	CA 1142190 A1	01-03-1983
		DK 35480 A	31-07-1980
		EP 0014020 A1	06-08-1980
		JP 55102084 A	04-08-1980
		JP 58026075 B	31-05-1983

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82