

(51) International Patent Classification:
G06F 15/16 (2006.01) **G06F 3/12** (2006.01)(21) International Application Number:
PCT/US2010/051025(22) International Filing Date:
1 October 2010 (01.10.2010)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant (for all designated States except US):
HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P. [US/US]; Hewlett-Packard Development Company, L.P., 11445 Compaq Center Drive West, Houston, Texas 77070 (US).

(72) Inventor; and

(71) Applicant : **ZENG, Jun** [US/US]; Hewlett-Packard Co., 1501 Page Mill Road, Palo Alto, California 94304 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **LIN, I-Jong** [US/US]; Hewlett-Packard Co., 1501 Page Mill Road,Palo Alto, California 94304 (US). **GIANNETTI, Fabio** [US/US]; Hewlett-Packard Co., 1501 Page Mill Road, Palo Alto, California 94304 (US).(74) Agents: **DAKIN, Lloyd** et al.; Hewlett-Packard Company, Intellectual Property Administration, 3404 E. Harmony Road, Mail Stop 35, Fort Collins, Colorado 80528 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG,

[Continued on next page]

(54) Title: GENERATING WORKFLOW SEQUENCES FOR PRINT JOBS

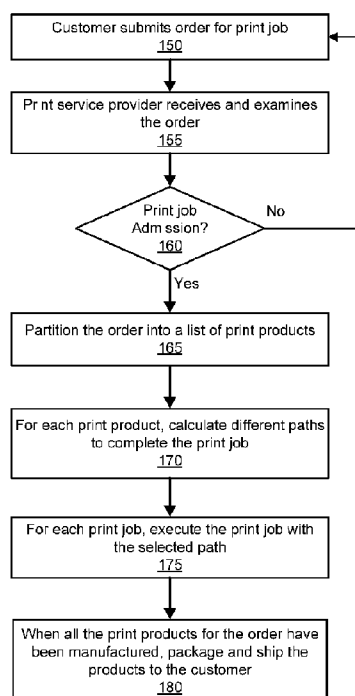


FIG. 1B

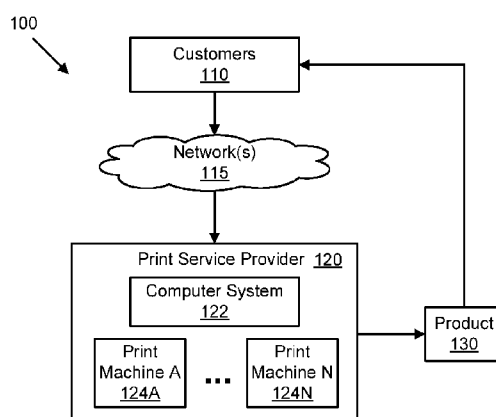


FIG. 1A

(57) Abstract: A method separates a print job into a print product and a sequence of tasks. The method estimates a time to setup a machine before executing a task and a time for the machine to execute the task. Machines are identified that have a potential bottleneck due to multiple concurrent printing requests being made to a machine. A workflow sequence is generated to complete the print job.



ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

- *as to the identity of the inventor (Rule 4.17(i))*
- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

- *with international search report (Art. 21(3))*

Declarations under Rule 4.17:

GENERATING WORKFLOW SEQUENCES FOR PRINT JOBS

FIELD OF THE INVENTION

5 The present invention relates to generating workflow sequences for print jobs.

BACKGROUND

10

Despite the “electronic age,” there is still demand for print services. Commercial print has annual retail sales over US\$700B. Print service providers (PSPs) fulfill the demand for print services by printing a variety of articles, such as photographs, brochures, course materials, periodicals and books,
15 advertisements, and product packaging.

20

The fulfillment of the print services is to convert print jobs of customers into printed products that are shipped to the customers. This process is complex and involves many different automated and manual steps that cannot be cost
effectively treated as a standard manufacturing activity. For instance, a print
production facility adapted to print and bind books includes many different types of machines and electronic equipment. This results in numerous combinations of factory configurations and workflows which may not be efficiently designed for optimal print performance.

25

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1A is a diagram of a print production system according to an example implementation.

5

Figure 1B is a flow diagram of the print production system of Figure 1A producing a product in accordance with an example implementation.

Figure 2 is a flow diagram for print production planning according to an example implementation.

10

Figure 3 is a diagram of a fulfillment process of a print product in which the fulfillment process is separated into a series of nine different tasks with a pre-defined precedence sequence according to an example implementation.

15

Figure 4 is an optimization algorithm to provide sequencing and assignment solutions for print production planning according to an example implementation.

Figure 5 shows a computer system in accordance with an example implementation.

20

SUMMARY OF THE INVENTION

One example embodiment is a method that separates a print job into a print product and a sequence of tasks. The method estimates a time to setup a machine before executing a task and a time for the machine to execute the task. Machines are identified that have a potential bottleneck due to multiple concurrent printing requests being made to a machine. A workflow sequence is generated to complete the print job.

DETAILED DESCRIPTION

Example implementations are systems, methods, and apparatuses for production planning, such as planning in print production and workflows. An example
5 embodiment generates an operational sequence of accepted jobs according to one or more business objectives (for instance, maximize service-level and profit) and assigns the sequence of the jobs to one or servers, printers, or machines. The jobs are classified, for instance, according to their due date tightness, and the servers are classified according to service demand. An optimization algorithm
10 that can be massively parallelized is implemented on a graphical process unit (GPU) to provide a solution in real-time. Example embodiments are implemented as a component of a work-flow solution or as a component of an operational planning tool, such as a print production planning tool.

15 One example embodiment relates to print manufacturing that is a make-to-order process. A print manufacturer or print service provider receives a print order of a customer and converts this order into a shipped print product, such as books, periodicals, calendars, stationeries, etc. Service level objectives (SLOs) for the print order include one or more of a completion time, costs for print production,
20 and other constraints. One business objective of the print service provider is to increase the throughput of print jobs to dilute fixed costs and maximize profit while satisfying SLOs and a desired quality of service.

Figure 1A is a diagram of a print production system 100 according to an example
25 implementation. Figure 1B is a flow diagram of the print production system 100 producing a product in accordance with an example implementation. These two diagrams are discussed together and illustrate an end-to-end workflow of a print manufacturing process for a print service provider 120.

30 According to block 150, a customer submits an order for a print job or submits a request for prints (e.g., submits a request for prints of a book).

For instance, one of the customers 110 provides the print job through one or more networks 115 to a print service provider 120. By way of illustration, the print job is provided through a store front (e.g., a customer physically visits a location to place the request or electronically submits the request through the internet).

According to block 155, the print service provider receives and examines the order for the print job.

According to block 160, a determination is made as to whether print job is accepted or admitted. If the print job is admitted, then flow proceeds to block 165. If the print job is not admitted, then flow proceeds back to block 150.

After the print job is received and accepted, a unique job ticket is assigned to the print job. The job ticket includes a job number, number of copies, due dates, prices and payments, client information, inks and presses, finishing, shipping methods, etc. Further stipulations can be provided as SLOs in a service level agreement (SLA).

Although not typically well-suited to an individual, the PSP may function to process print jobs for multiple individuals, such as, the customers of a large retailer, wherein the large retailer takes orders from the individuals (e.g., for photo calendars) and submits an order as a batch of individual customer orders to the PSP. For example, the customer is a large retailer submitting the order on behalf of many individuals. Of course the systems and methods described herein are not limited to any particular type or size of customer or customers, and may also be utilized with individual customers of the PSP. In general, the customer creates the material to be printed (e.g., the photographs and brochures, course materials, periodicals and books, to advertisements and product packaging) or works with a third-party provider to generate the material to be printed. The customer then submits an order including one or more materials for the PSP to

print, along with one or more print parameters (e.g., substrate stock, number of copies, due date, and any special instructions such as laminating and quality level, and shipping information).

- 5 The submitted print job (e.g., submitted as electronic files in the form of Portable Document Format (PDF) and/or a Tagged Image File Format (TIFF)) are examined for correctness (preflight), edited for color quality and accuracy, imposed, and flattened and half-toned (raster imaging process) to create
10 bitmaps.

According to block 165, the accepted order is partitioned into a list of different print products. A "print job" typically include, but is not limited to, a list of items to instruct the print operations, for instance, book titles, number of copies, and some or all of the print parameters from the order, and an expected due date. In
15 addition, the PSP may assign additional priority information, for instance, the characterization of the customer, e.g., a repeat and high-volume customer may receive a higher priority from the PSP 100 than a first-time or low-volume customer. The print job may also include other parameters assigned by the PSP, for example, based on current backlog, supplies in stock, and so forth.

An order can be a Job Definition Format (JDF) file initiated during a content creation stage. The order can also be in other forms of Extensible Markup Language (XML), Personalized Print Markup Language (PPML) file format, or another format negotiated between the print service provider and their upstream
25 digital content provider. The order describes complete fulfillment specifications in addition to other information. These specifications are then used in a print manufacturing environment of the print service provider 120 to produce and ship the final printed product. Customer information includes shipping and payment and provides a financial measurement of late penalty (e.g., costs to upgrade a
30 shipping method to compensate for a manufacturing delay).

According to block 170, for each print product, different or alternate paths are calculated to complete the print job. For example, the print job request is analyzed by a computer system 122 to find different workflow paths for completing the print job at the print manufacturing facility.

5

One example embodiment provides quantitative prediction, analysis, and operational-level assistance for the print manufacturing process. This includes determining different production paths through the print manufacturing facility to produce the product. For example, one example embodiment narrows the fulfillment to a handful of product types that contribute to revenue according to past order history and marketing forecast. A list of product types is then used to guide the optimization of the factory floor and order admission. Each product type has its pre-defined fulfillment path; resources are shared among different fulfillment paths when optimal. The fulfillment paths are further augmented with a list of resources that map the tasks demanded by the fulfillment paths with one or more of software, automated machines (e.g. printing machines), operator-assisted machines, hardware, and operators that carry out the respective tasks.

Multiple orders released to the factory floor for simultaneous production compete for resources. Priorities for services are provided to urgent orders (e.g., priority is based on a deadline specified in a SLA). An order sequence is calculated based on the priorities. Similarly there may be multiple machines that can fulfill the same task. Here, priorities for the task assignment are given to the most effective machines orders (e.g., calculate the effectiveness of a printing machine according to the operations objective and attributes and states of the printing machine).

According to block 175, each print job is executed with a selected one of the paths or workflows. Here, the print job is sent to one or more servers or print press or finishing machines 124A – 124N. For example, printer-ready electronic files are sent to print presses to produce physical copies. Printed sheets or web

are divided into printed pages that are then folded, collated, and bound into book blocks, if the final product is a book. The book blocks are joined to book covers. The print machines 124A-124N refer to all machinery, resources, and procedures used to complete the print product, including print presses, cutters, laminators,
5 binders, shipping stations, and more.

According to block 180, a final product 130 is shipped from the print service provider 120 to the customer 110 once all the print products within the same order have been collected. For example, if the print products are books, then the
10 finished books are sorted. When all the printed books for this order are collected, they are packaged together. The package is labeled according to shipping instruction, for instance, shipped by Federal Express or U. S. Post Office. The labeled package is then placed at the shipping station for the shipper (for instance, DHL, Federal Express) to pick up.

15 Figure 2 is a flow diagram for print production planning according to an example implementation. One or more of these blocks can be performed by the print service provider 120 shown in Figure 1.

20 According to block 200, for each admitted job, the job is separated into a list of print products. For example, an admitted order may include two copies of three different book titles therefore six print products. Next, the manufacturing procedure of each print product is separated into a sequence of tasks, such as raster image processing (RIP), black and white printing, case binding, kitting,
25 shipping, etc.

As used herein, a task is defined as work that can be completely fulfilled by a unit of one server type, for instance, one print press out of the array of print presses on the factory floor. For each job, the sequence of tasks has precedence
30 constraints that certain tasks must be executed or carried out before other tasks, for instance, in FIG. 3, RIP 310 must be accomplished before prints 320, 330.

This task sequence can be predefined by a set of workflow templates, for instance, transactional, periodical, books, photobooks, publications, and then the accepted job is then assigned to the corresponding workflow template.

- 5 According to block 205, for each print product, the method defines a fulfillment path (i.e., a sequence of tasks).

Figure 3 is a diagram of a fulfillment process or a print production job 300 of a print product in which the fulfillment process is separated into a series of nine
10 different tasks with a pre-defined precedence sequence according to an example implementation. For example, the print production job creates a book having black and white printed pages with a color cover.

Raster image processing 310 generates a raster image or bitmap that is sent to a
15 black and white printer 320 and a color printer 330. The black and white printer 320 prints the internal pages of the book, and the color printer 330 prints the cover and/or color pages. At block 340, slit, cut, and collating operations occur, and at block 350, the block is bound (i.e., makes part of the book block). At block 370, glue operations occur (e.g., pages of the book are glued together at the
20 binding). At block 380, trim operations occur. At block 390, this book is packaged, labeled and mail-sorted.

Turning back to Figure 2, according to block 210, the method estimates for each task both the setup time and the work time for available servers (e.g., all servers
25 able to complete this particular task). For example, the setup time is sequence dependent (i.e. the setup time depends on the characteristic of the task processed by this server prior to this task). To account for this sequence dependency, a common baseline is defined for similar tasks or a same family of tasks. For each task, the recorded setup time is that time that assumes the
30 baseline is the task processed in a prior print job. For instance, assuming the setup times for tasks A, B, C are 5, 15, 10 measured from the baseline, then for

job sequence {ABC}, the setup cost is {5,10,5}; for job sequence {ACB}, the setup cost is {5,5,5}. This job sequence minimizes the setup cost for the given tasks A, B and C. Additionally, the due date tightness (e.g., SLACK time, defined as (due date – work time – release date)) for each task is estimated.

5

One example embodiment estimates the workload (e.g., work that can be completed by one print production machine). For example, this estimate includes the workload that can be completed by plural different printing machines that can be used to complete the print job. This workload includes both variable costs (e.g., the number of pages to be printed) and fixed costs (for example, setup cost, e.g., cost to load a paper roll in a printing machine). Reducing the fixed cost helps to improve the manufacturing throughput and reduced the lead time, which increases customer satisfaction. For example, assume a particular job to be printed requires printing on a particular substrate, for instance, cream paper.

10

Further assume that machine press A already has the cream paper roll loaded, but machine press B has the white paper roll loaded. One example embodiment chooses machine press A over machine press B so the job can be completed without the overhead of offloading and loading paper roll. If press B were chosen, a person would need to unload the currently-loaded white paper roll, and then reload the cream paper roll. Thus, one embodiment estimates a fixed cost that includes determining whether the print job can be started and completed by the printing machine without loading paper into the print press.

20

According to block 220, the method classifies the servers into server types and identifies bottleneck server types. Here servers refer to a software server, a machine, or a worker that can completely fulfill a single task. For example, the print presses that are capable of printing black-and-white pages are grouped into one server type. The utilization rate of each sever type represents the ratio of the demand for the service of this particular server type over the capacity of this particular server type. For instance, this utilization rate can be determined as the ratio of the busy time over total available working time during a particular time

interval, for instance, a week. The busy times and the total available working times can be extracted from past historical factory record. The server type holds highest utilization rate is called the resource bottleneck. That is, this particular server type is the scarcest resource of the entire print manufacturing workflow because its service demand is closest to its maximum capacity. All the server types are placed into a server type queue. This server type queue is sorted according to the utilization rate in the descendent order. Therefore, the first element in this server type queue has the highest utilization rate therefore the resource bottleneck.

The task sequencing and assignment procedure starts with the resource bottleneck server type. Once all the assignment is done with this resource bottleneck, this server type will be deleted from the server type queue. The first element of current server type queue is the bottleneck among all yet-to-be-scheduled resource types. We now schedule the tasks of this service type. Once the scheduling is done, this server type will be deleted from the server type queue. We repeat this procedure until the server type queue is empty, that is, all the server types have been scheduled. This way the degree of usage optimization of server types are aligned with their order of the scarceness.

By way of example, consider a print factory that produces only booklets. The fulfillment path starts with RIP, followed by PRINT, BINDING and MAILING. There are 2 RIP servers, 2 print presses, 6 binders of different generations, and one mail-sort station. Therefore there are four server types. For instance, all 6 binders form one "binder" server type. Historical record shows 70% of the time the two print presses are busy printing pages. Therefore, the utilization rate for the "print" server type is 70%. Historical record also shows RIP servers 50% of utilization rate, binder servers 40% of utilization rate, and the mail-sort station 45% utilization rate. Therefore, the server type queue will consist of PRINT, RIP, MAIL, and BINDER, following the descendent order of the corresponding utilization rate. PRINT will be the bottleneck to be first scheduled, and it is

removed from the server type queue afterwards. Next is to schedule the first element of the remaining server type queue, which is RIP; repeat for MAIL; and perform BINDER last. It should be noted that the servers within the same server type may have different capacity, capability and availability even though they can fulfill the same task. For instance, the 6 binders are of different generations, some can be newer than others and hence have different performance characteristics (e.g., binder A is newer than binder B and binds more booklets per minute). With block 220, example embodiments classify each server on the floor of the factory. This process is a resource classification procedure.

For each server type, according to block 230, its corresponding tasks are classified. Task classification is according to the task's priority. The priority of a task can be calculated according to its lead time, its due date, and other operating parameters. In one embodiment, the priority of a task equals to its due
5 date. In another embodiment, the priority of a task equals to the difference between the due date and the lead time (that is, time required to complete this task). In another embodiment, the PSP assigns all jobs from a repeat and high-volume customer an additional weight factor to elevate their priority. In another embodiment, the PSP assigns all high-value jobs (jobs with higher margin rate)
10 an additional weight factor to elevate their priority. Tasks according to their priority values are assigned to different task bins. The tasks within the highest priority bin are scheduled first, and then the tasks within the second highest priority bin. This is repeated until all the tasks are scheduled.

15 According to block 240, an optimization algorithm is applied to generate a sequence for the task and assignment solutions.

For the selected server type (performed at block 220, there may be multiple servers that can complete the same task but have different capability, capacity,
20 and availability) and the corresponding selected task class (performed at block 230, there may be multiple tasks competing the same type of resources), an

optimization algorithm is applied to generate the assignment of tasks to each server within this server type, and the order that tasks will be performed by each server.

- 5 By way of example, consider the aforementioned print factory that produces booklets. There are 20 binding tasks to be assigned to the binder server type with 6 binders. These 20 binding tasks can be partitioned into three task bins according to their priority value, Bin A, B and C. Tasks in bin A belong to the highest-priority category; tasks in bin C belong to the lowest-priority category.
- 10 The assignment starts with assigning all tasks in bin A to all the 6 binders. In one embodiment, this assignment is carried out by an optimization algorithm to maximize the operations objective, for instance, load balancing among different servers, or, minimize lead time for tasks of high priority. Next, this optimization algorithm creates assignment solutions for tasks in bin B. Following that is the
- 15 assignment for tasks in bin C.

One example embodiment finds task sequences and binding tasks with an optimal choice of the candidate servers. Brutal-force optimization techniques (e.g., mixed integer programming) can be applied; but they may not be able to

20 satisfy the real-time constraints (in-line solution) due to the hyper polynomial computational cost. As such, one embodiment executes an algorithm that is a hybrid of a Genetic Algorithm and Simulated Annealing that provides massive parallelizability so that it can be implemented on graphical process unit (GPU) architecture to deliver real-time performance. This hybrid also overcomes the

25 traps of local minima/maxima thanks to the features adapted from simulated annealing.

One example embodiment is a method that obtains a print job that requires plural different servers to execute. The method estimates times for different sequences of the servers to execute the print job and identifies potential bottlenecks among all sever types involved in the workflow. The method then analyzes the times for

the different sequences and the potential bottlenecks and selects one of the different sequences to execute the print job. For example, an optimal sequence is selected (e.g., one having the optimal lead time distribution for all the print jobs, that is, achieving the maximal on-time delivery).

Figure 4 shows an optimization algorithm 400 to provide sequencing and assignment solutions for print production planning.

- 5 One example embodiment executes a Parallel Recombinative Simulated Annealing (PRSA) algorithm. A PRSA fitness function is a weighted combination of multiple objectives, including takt time (i.e., maximum time per unit to produce a product to meet a specified demand), makespan (i.e., the time difference between the start and finish of a sequence of jobs or tasks), service level, cost-
10 per-page, etc.

- An alternative embodiment is to eliminate the outer loop (block 220 in Figure 2) to simplify the calculation. For the bottleneck sever type, the task sequencing and binding is carried out. For all other server types, the task sequence is determined
15 such that the jobs maintain the same order as the bottle neck server type.

- Example embodiments provide methods and apparatus that enable the delivery of in-line planning solutions that are applicable to both job-shop and flow-shop configurations. As a component of a planning tool, example embodiments identify
20 additional productivity and optimal job mixes that maximize profit subject to acceptable service levels and help to allocate resource investment strategies that maximize return. As a component of an in-line production management tool, example embodiments provide real-time sequencing and resource binding solutions for accepted jobs to achieve (pareto)-maximum of service-level and
25 profit.

Example embodiments are implemented in various apparatus and methods that

provide operational level planning assistance accounting for job assignment and sequencing, such as being a component of print production planning software or a component of a work-flow solution.

- 5 Figure 5 is a block diagram of a computer system 500 in accordance with an exemplary embodiment of the present invention. By way of example, the computer system is implemented by a print service provider or in a print production facility.
- 10 In one embodiment, the computer system includes a database or warehouse 560 and a computer or electronic device 505 that includes memory 510, algorithms and/or computer instructions 520, display 530, processing unit 540, and one or more buses 550.
- 15 In one embodiment, the processor unit includes a processor (such as a central processing unit, CPU, multi-core CPU, microprocessor, GPU, application-specific integrated circuit (ASIC), etc.) for controlling the overall operation of memory 510 (such as random access memory (RAM) for temporary data storage, read only memory (ROM) for permanent data storage, and firmware). The processing unit
- 20 540 communicates with memory 510 and algorithms 520 via one or more buses 550 and performs operations and tasks necessary for constructing models and searching the database per a query. The memory 510, for example, stores applications, data, programs, algorithms (including software to implement or assist in implementing embodiments in accordance with the present invention)
- 25 and other data.

In one example embodiment, one or more blocks or steps discussed herein are automated. In other words, apparatus, systems, and methods occur automatically. The terms “automated” or “automatically” (and like variations thereof) mean

30 controlled operation of an apparatus, system, and/or process using computers

and/or mechanical/electrical devices without the necessity of human intervention, observation, effort and/or decision.

The methods in accordance with example embodiments are provided as
5 examples and should not be construed to limit other embodiments within the scope of the invention. Further, methods or steps discussed within different figures can be added to or exchanged with methods of steps in other figures. Further yet, specific numerical data values (such as specific quantities, numbers, categories, etc.) or other specific information should be interpreted as illustrative
10 for discussing example embodiments. Such specific information is not provided to limit example embodiments.

In some example embodiments, the methods illustrated herein and data and instructions associated therewith are stored in respective storage devices, which
15 are implemented as one or more computer-readable or computer-usable storage media or mediums. The storage media include different forms of memory including semiconductor memory devices such as DRAM, or SRAM, Erasable and Programmable Read-Only Memories (EPROMs), Electrically Erasable and Programmable Read-Only Memories (EEPROMs) and flash memories; magnetic
20 disks such as fixed, floppy and removable disks; other magnetic media including tape; and optical media such as Compact Disks (CDs) or Digital Versatile Disks (DVDs). Note that the instructions of the software discussed above can be provided on one computer-readable or computer-usable storage medium, or alternatively, can be provided on multiple computer-readable or computer-usable
25 storage media distributed in a large system having possibly plural nodes. Such computer-readable or computer-usable storage medium or media is (are) considered to be part of an article (or article of manufacture). An article or article of manufacture can refer to any manufactured single component or multiple components.

30

Example embodiments are implemented as a method, system, and/or apparatus. As one example, example embodiments and steps associated therewith are implemented as one or more computer software programs to implement the methods described herein. The software is implemented as one or more modules
5 (also referred to as code subroutines, or "objects" in object-oriented programming). The software programming code, for example, is accessed by a processor or processors of the computer or server from long-term storage media of some type, such as a CD-ROM drive or hard drive. The software programming code is embodied or stored on any of a variety of known physical and tangible
10 media for use with a data processing system or in any memory device such as semiconductor, magnetic and optical devices, including a disk, hard drive, CD-ROM, ROM, etc. The code is distributed on such media, or is distributed to users from the memory or storage of one computer system over a network of some type to other computer systems for use by users of such other systems.
15 Alternatively, the programming code is embodied in the memory and accessed by the processor using the bus. The techniques and methods for embodying software programming code in memory, on physical media, and/or distributing software code via networks are well known and will not be further discussed herein.

20 The above discussion is meant to be illustrative of the principles of various example embodiments. Numerous variations and modifications will become apparent to those skilled in the art once the above disclosure is fully appreciated. It is intended that the following claims be interpreted to embrace all such
25 variations and modifications.

CLAIMS

What is claimed is:

- 1) A method executed by a computer, comprising:
 - separating, by the computer, a print job into print products;
 - separating, by the computer, each of the print products into a sequence of tasks;
 - estimating, by the computer and for multiple machines capable of completing the tasks, a time to setup a machine before executing a task and a time for the machine to execute the task;
 - identifying, by the computer, machines having a potential bottleneck due to multiple concurrent printing requests being made to one of the machines;
 - and
 - generating a workflow sequence to complete the print job.
- 2) The method of claim 1, wherein the time to setup a machine is derived from a common baseline that defines similar tasks previously processed by the machine for a prior print job.
- 3) The method of claim 1 further comprising, estimating a workload that can be completed by plural different servers used to complete the sequence of tasks.
- 4) The method of claim 1 further comprising, estimating a fixed cost that includes determining whether the print job can be started and completed by a printing machine without loading paper into the printing machine.
- 5) The method of claim 1, wherein the print job is an order to produce books, and an optimization algorithm selects a sequence of the machines to print pages of the books, fold and glue bindings of the books, and assembly and bind the print pages into the books.

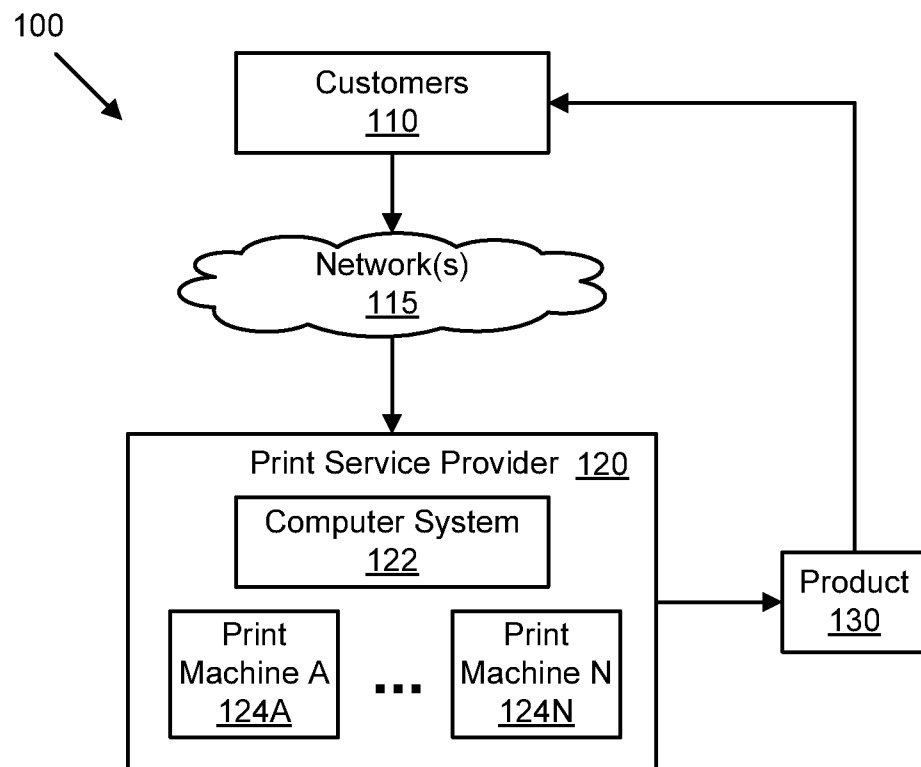
- 6) A tangible computer readable storage medium storing instructions that, when executed by a computer, cause the computer to:
- obtain print jobs that require plural different servers to execute;
 - separate each of the print jobs into a list of print products and each of the print products into a sequence of tasks;
 - estimate, for each of the tasks, times for different sequences of the servers to execute the print jobs;
 - identifying potential bottlenecks at each of the servers according to usage demand in reference to capacity; and
 - analyze the times for the different sequences and the potential bottlenecks to select one of the different sequences to execute the print jobs.
- 7) The tangible computer readable storage medium of claim 6 wherein the computer further executes the instructions:
- calculate, for each task to be assigned to a bottleneck server type, a priority value that is based upon one of due time, lead time, and preferential customer;
 - categorize the tasks into different priority classes;
 - apply an optimization algorithm to generate an optimal assignment of the tasks to the servers within a server type; and
 - produce an end-to-end production sequence and assignment for a print service provider for the print jobs to be fulfilled.
- 8) The tangible computer readable storage medium of claim 6 wherein the computer further executes the instructions to rank the print jobs according to due dates provided in a service level agreement (SLA).
- 9) The tangible computer readable storage medium of claim 6, wherein the computer further executes the instructions to execute a Parallel

Recombinative Simulated Annealing (PRSA) algorithm to select a task assignment solution to execute the tasks.

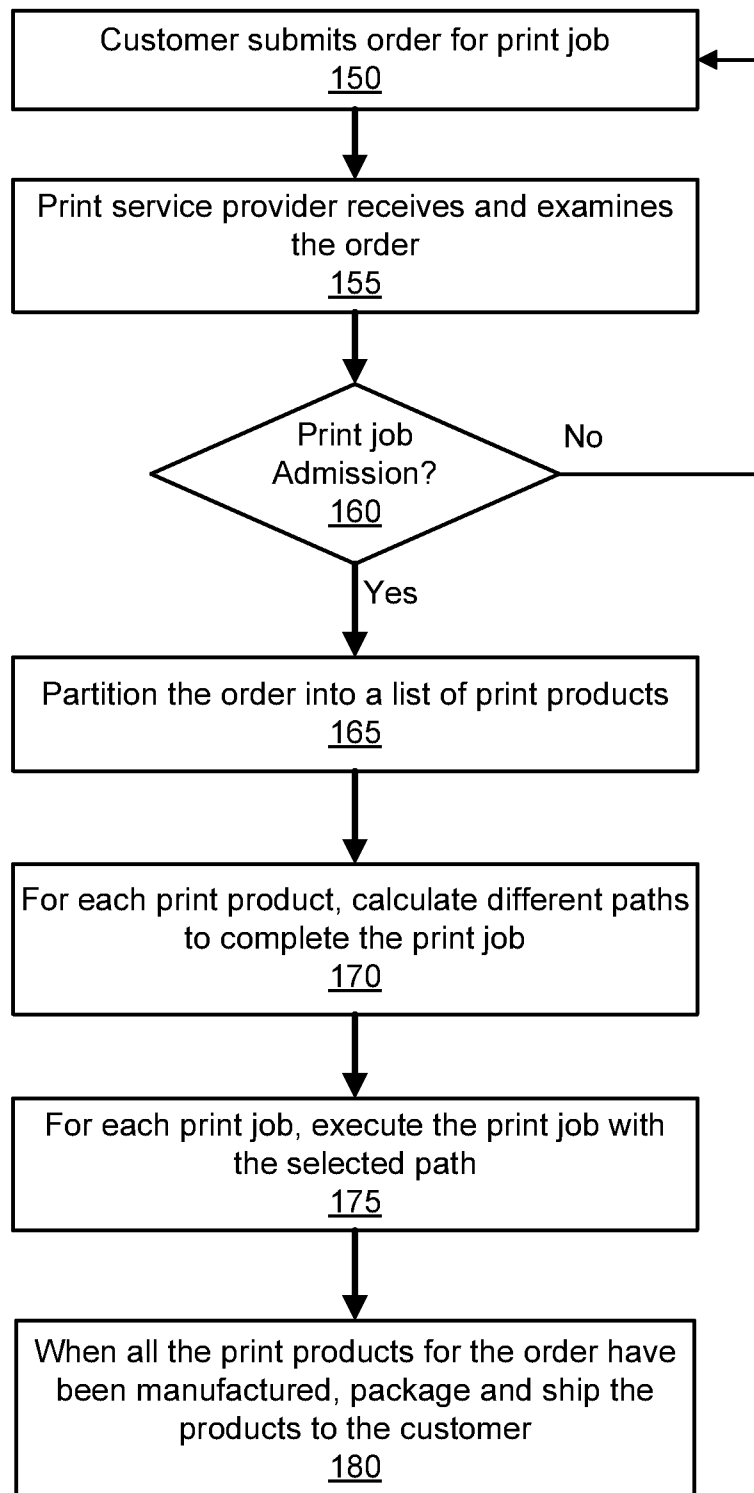
- 10) The tangible computer readable storage medium of claim 6, wherein the computer further executes the instructions to estimate, for each of the plural different servers, a time to setup a server before executing the print jobs and a time to complete the print jobs.
- 11) A computer, comprising:
 - a memory storing instructions; and
 - a processor that executes the instructions to:
 - retrieve print jobs that require plural different servers to execute;
 - separate each of the print jobs into a list of print products;
 - separate each of the print products into a sequence of tasks;
 - estimate, for each of the tasks, a time to setup a server before executing a task and a time for the server to execute the task for servers that are capable of completing the task;
 - identifying potential bottlenecks at each of the servers according to usage demand in reference to capacity; and
 - analyze setup times at the servers, completion times at the servers, and the potential bottlenecks to select a sequence of servers to execute the print jobs.
- 12) The computer of claim 11 wherein the processor further executes the instructions to consider multiple objectives that include takt time at the servers, makespan, cost-per-page to print pages of the print jobs.
- 13) The computer of claim 11, wherein the print jobs are a request to print produce copies of a book.

- 14) The computer of claim 11 wherein the sequence of tasks include raster image processing (RIP), black and white printing, color printing, block binding, and gluing pages to a binding.
- 15) The computer of claim 11 wherein the processor further executes the instructions to estimate a time for each of the servers to execute a portion of the print jobs.

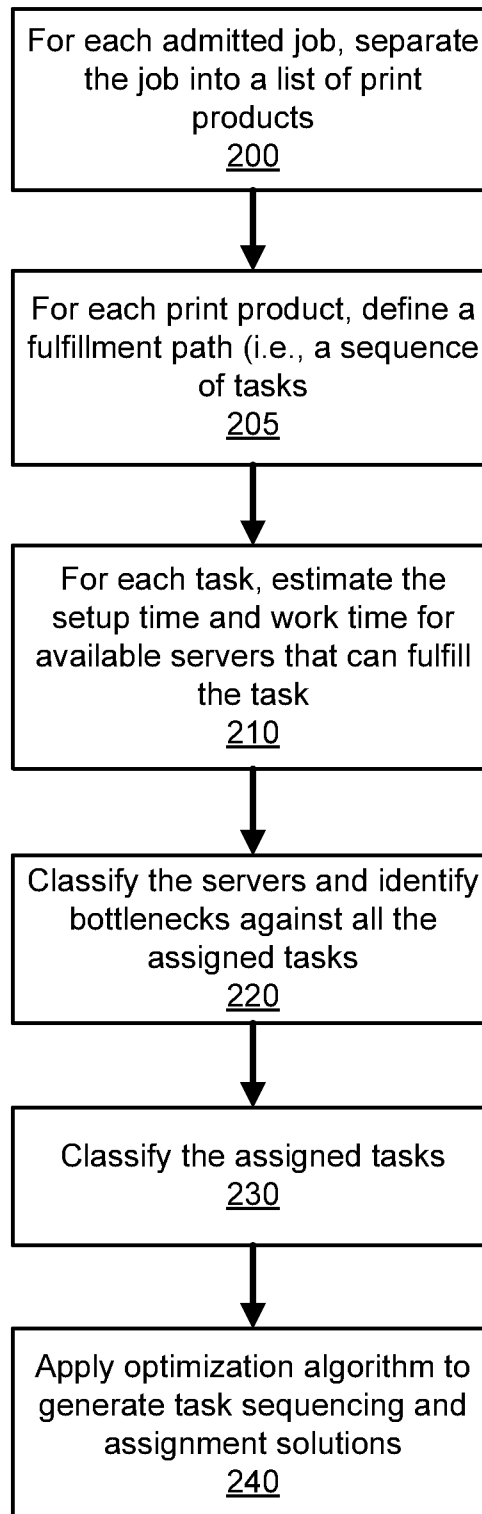
1/6

**FIG. 1A**

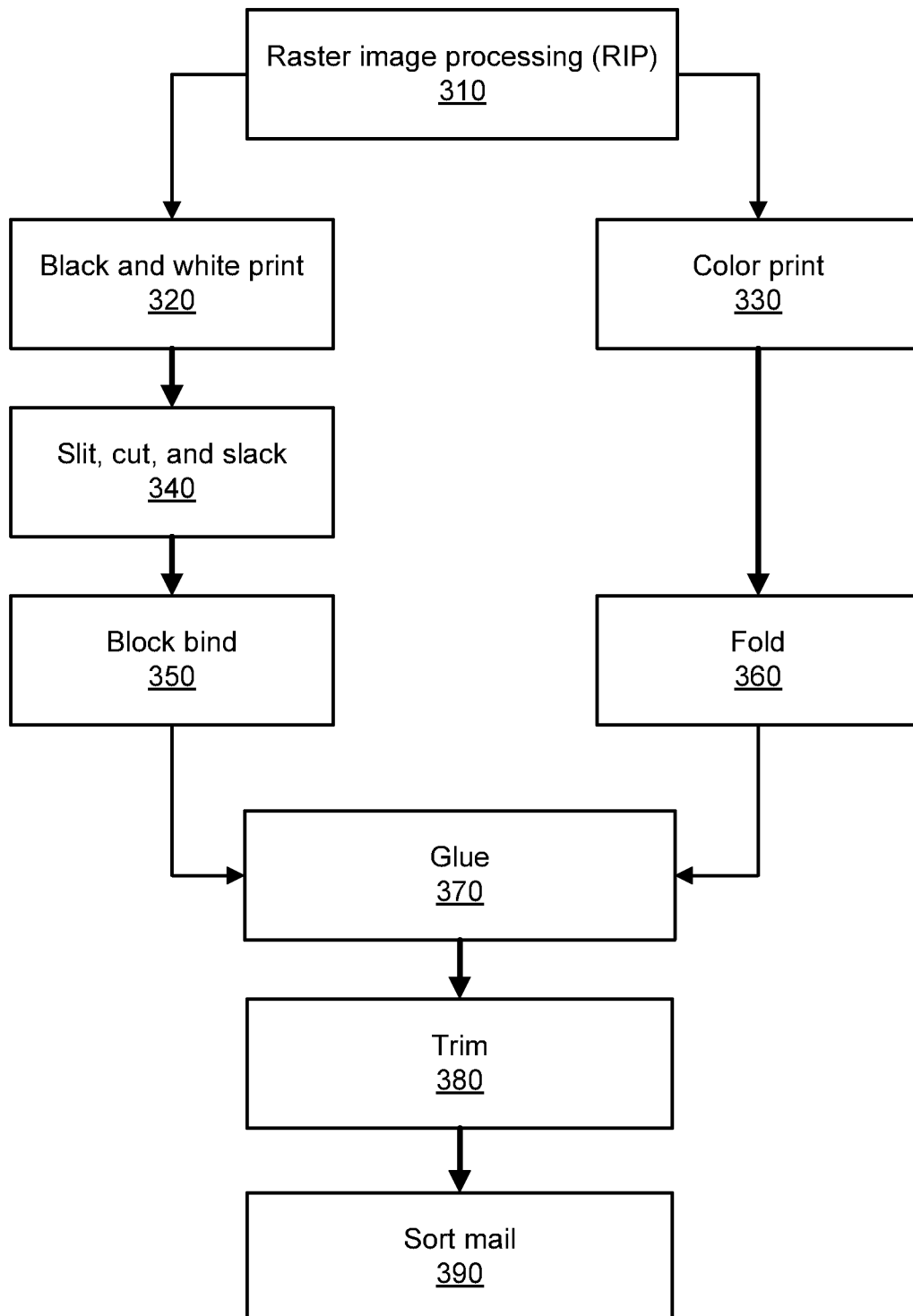
2/6

**FIG. 1B**

3/6

**Fig. 2**

4/6

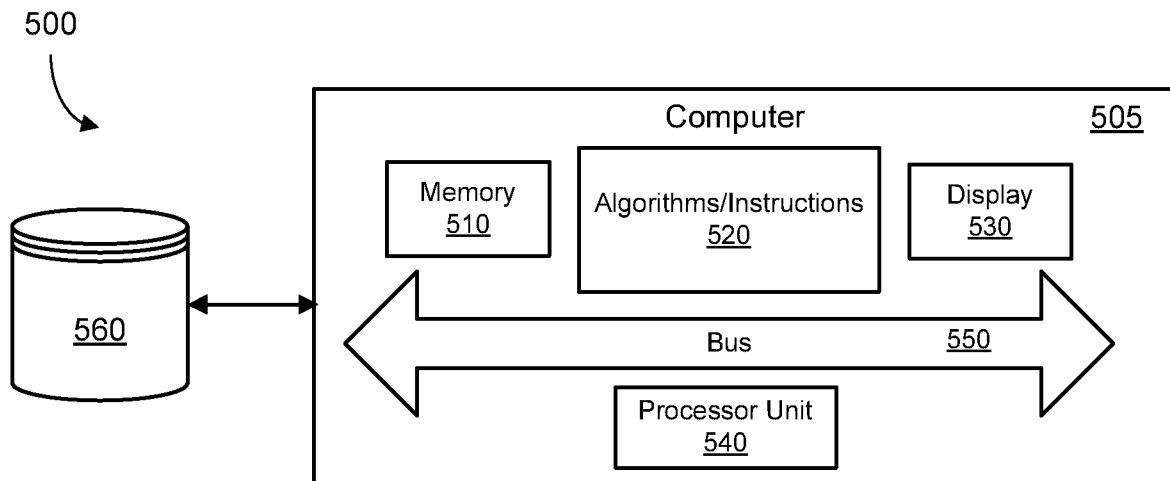
**Fig. 3**

5/6

```
1 /* Initialization */
2 Set initial population of chromosome
3 Implement the high-level synthesis by
4 using the information of initial chromosome:
5   Phase I: Assign tasks to servers
6   Phase II: Sequencing tasks according to their priorities
7 Set the initial temperature T
8
9 while (Stopping criteria of annealing is not satisfied)
10   for i = 1: N /* Inner loop of annealing process */
11     /* Evolution strategy */
12     Reproduction:
13       Best chromosomes copied to the next generation;
14     Crossover:
15       Parameterized uniform crossover is employed
16       to generate the child chromosome
17       from two randomly-selected parent chromosomes;
18     Boltzmann trial:
19       if Fitness(child) < Fitness(parents) or
20         rand(0, 1) < exp(-[Fitness(child)-Fitness(parents)]/T):
21         Child chromosome is selected;
22       else:
23         Parent chromosome (the best one) is selected;
24       end if
25     Mutation:
26       New chromosomes are generated randomly;
27     New population replaces the old generation
28   end for
29
30   T= rate × T
31 end while
32
33 Optimum chromosome:
34   find the chromosome within the final population
35   with the best fitness value;
36 Output the results of task sequencing and server assignment.
```

**Fig. 4**

6/6

**Fig. 5**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2010/051025**A. CLASSIFICATION OF SUBJECT MATTER****G06F 15/16(2006.01)i, G06F 3/12(2006.01)i**

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 15/16; G06F 15/00; G06F 3/12

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) & Keywords: print, job, load, multiple machines, separating

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 20030184799 A1 (FERLITSCH, A. R.) 02 October 2003 See abstract, paragraphs [12]–[22], and claims 1–21.	1–15
A	US 20070019228 A1 (RAI, S. et al.) 25 January 2007 See abstract, figure 1, and claims 1–25.	1–15

☐ Further documents are listed in the continuation of Box C.☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

24 JUNE 2011 (24.06.2011)

Date of mailing of the international search report

27 JUNE 2011 (27.06.2011)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
Government Complex-Daejeon, 189 Cheongsu-ro,
Seo-gu, Daejeon 302-701, Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

AHN, Cheol Yong

Telephone No. 82-42-481-8371



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2010/051025

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 20030184799 A1	02.10.2003	JP 04-509965 B2	14.05.2010
		JP 2003-241948 A	29.08.2003
		JP 2003-271349 A	26.09.2003
		JP 2006-319974 A	24.11.2006
		US 2002-0089687 A1	11.07.2002
		US 2002-0089688 A1	11.07.2002
		US 2002-0089689 A1	11.07.2002
		US 2002-0089691 A1	11.07.2002
		US 2002-0097424 A1	25.07.2002
		US 2002-0097429 A1	25.07.2002
		US 2002-0113989 A1	22.08.2002
		US 2002-0114004 A1	22.08.2002
		US 2003-0007177 A1	09.01.2003
		US 2003-0007179 A1	09.01.2003
		US 2003-0227644 A1	11.12.2003
		US 2004-0061892 A1	01.04.2004
		US 2005-0206917 A1	22.09.2005
		US 2007-0127063 A1	07.06.2007
		US 7046384 B2	16.05.2006
		US 7180619 B2	20.02.2007
		US 7190477 B2	13.03.2007
		US 7265860 B2	04.09.2007
		US 7339687 B2	04.03.2008
		US 7480068 B2	20.01.2009
		US 7548331 B2	16.06.2009
		US 7679770 B2	16.03.2010
		US 7706017 B2	27.04.2010
US 20070019228 A1	25.01.2007	None	