



Vaasan yliopisto  
UNIVERSITY OF VAASA

OSUVA Open  
Science

This is a self-archived – parallel published version of this article in the publication archive of the University of Vaasa. It might differ from the original.

## Cloud manufacturing – scheduling as a service for sheet metal manufacturing

**Author(s):** Helo, Petri; Phuong, Duy; Hao, Yuqiuge

**Title:** Cloud manufacturing – scheduling as a service for sheet metal manufacturing

**Year:** 2018

**Version:** Accepted manuscript

**Copyright** © 2018 Elsevier. Creative Commons Attribution–NonCommercial–NoDerivatives 4.0 International (CC BY–NC–ND 4.0) license, <https://creativecommons.org/licenses/by-nc-nd/4.0/>

### Please cite the original version:

Helo, P., Phuong, D., & Hao, Y., (2018). Cloud manufacturing – scheduling as a service for sheet metal manufacturing. *Computers and operations research* 110, 208–219. <https://doi.org/10.1016/j.cor.2018.06.002>

# Cloud manufacturing – scheduling as a service for sheet metal manufacturing

---

## **Abstract**

Cloud manufacturing refers to a new concept of using centralized cloud computing for manufacturing information systems to support distributed and dynamic collaborative manufacturing environment. The core of cloud manufacturing is to provide service to geographically distributed manufacturers centralized services. This paper introduces a cloud based production scheduling system for sheet metal manufacturing and discusses the requirements of scheduling as a service. A cloud-based genetic algorithm based scheduling application has been developed to serve distributed manufacturing. The characteristics of the prototype system are described and performance estimates are tested. The solution shows how scheduling as a service can be built as part of connected smart machinery and provide a generic platform for varying uses.

**Keywords:** cloud manufacturing, sheet metal, flow shop, scheduling, genetic algorithm.

## **1. Introduction**

Cloud manufacturing is a novel concept referring to operating manufacturing related software in cloud infrastructure or in a broader sense referring to virtual factories, i.e. supply chains that carry the production and operations in multiple locations and providing close to real-time information from the events (Tao et al., 2011; Xu, 2012). Both approaches acknowledge the importance of application of Internet technology. By using the communication technology, the up to date information can be shared between the machines, production lines, factories and different actors within the supply chains. Some authors, such as Wu et al. (2015) see a new manufacturing

paradigm emerging from this technology to deal with the competitive and dynamic manufacturing environment. Visions of future manufacturing and its associated manufacturing system have been presented (Wu et al., 2013). It is vital to provide higher level of operational performance in the manufacturing system to achieve green technology and sustainable operations (Park & Jeong, 2013).

Cloud computing is considered as green IT technology, because it can provide the computing resources as services and enable rapid provisions with minimal management effort (Park & Jeong, 2013). Cloud computing reduces the environmental impact of computing resources and improve the efficiency of computing resources. By using cloud-enabled architecture for manufacturing systems, the product lines can be monitored even they are in distributed locations, while guarantee the operational performance of systems.

Part of the cloud manufacturing concept is about servitization – this means that manufacturing is gradually transferring from traditional product-oriented type to service-oriented type (Huang et al., 2013; Luo et al., 2011). Any manufacturing resources and capabilities are provided as services. The characteristics of new era of manufacturing are agile, networked, sustainable and sharing information with a temporary collaboration nature. A number of similar paradigms have emerged in parallel, such as distributed manufacturing, dispersed network manufacturing, and virtual enterprises (Ford et al., 2012). Each manufacturing technology or model has its own emphasis and has played an important role in the development of manufacturing industries (Zhang et al., 2014) and they can be classified into two different aspects: structure oriented concepts which mainly focus on the business formation and technology oriented concepts which emphasize the technology implementation (Hao & Helo, 2014). Sharing information in the supply chain is an important feature of many cloud manufacturing approaches.

In cloud manufacturing environment, the manufacturing tasks are parallel operated in different production lines (Zhang et al., 2014). The scheduling mechanism centrally managed. Fujii, Kaihara & Morita (2000) design the integrated simulation model to endorse different criteria in virtual factory context. Similar integrated approach is proposed in cloud manufacturing. Shared centralized data repository and shared computing resource is another type of approach on cloud manufacturing. Production scheduling is a computational task needed to improve the efficiency of operations and resource utilization. The scheduling task can be divided into three main phases: (1)

problem formulation, (2) optimization task and (3) evaluation of results. Problem formulation and evaluation of results are conducted locally by the production management. The actual optimization task, proposing a suitable sequence of jobs, is a part of process which may be conducted at a centralized scheduling system and computing resource may serve several users.

Targeting the cloud manufacturing, the purpose of this paper is to demonstrate a prototype of a cloud system to support the scheduling task in production planning and control in the context of sheet metal manufacturing environment. As many of the production planning and control techniques in sheet metal manufacturing are relaying on optimization or other computing intensive mathematical operations, there is a potential of using cloud computing architecture to take the full utilization of the line. In order to test the concept of cloud manufacturing in production planning and control, a prototype system was developed and tested. The developed system enables real-time scheduling as a service in the flow shop scheduling problem. The objectives of this research are as follows:

- (1) To demonstrate a new approach model to provide scheduling as a service to specific type of sheet metal production lines which are in distributed locations and are centrally managed.
- (2) A GA based approach to solve multi-objective problems is proposed. The centralized cloud solution can separate the optimization problem and solve the problem with different principles.

The reminder of this paper follows with literature review, a short introduction on the developed prototype model, the actual results and finally a conclusions section.

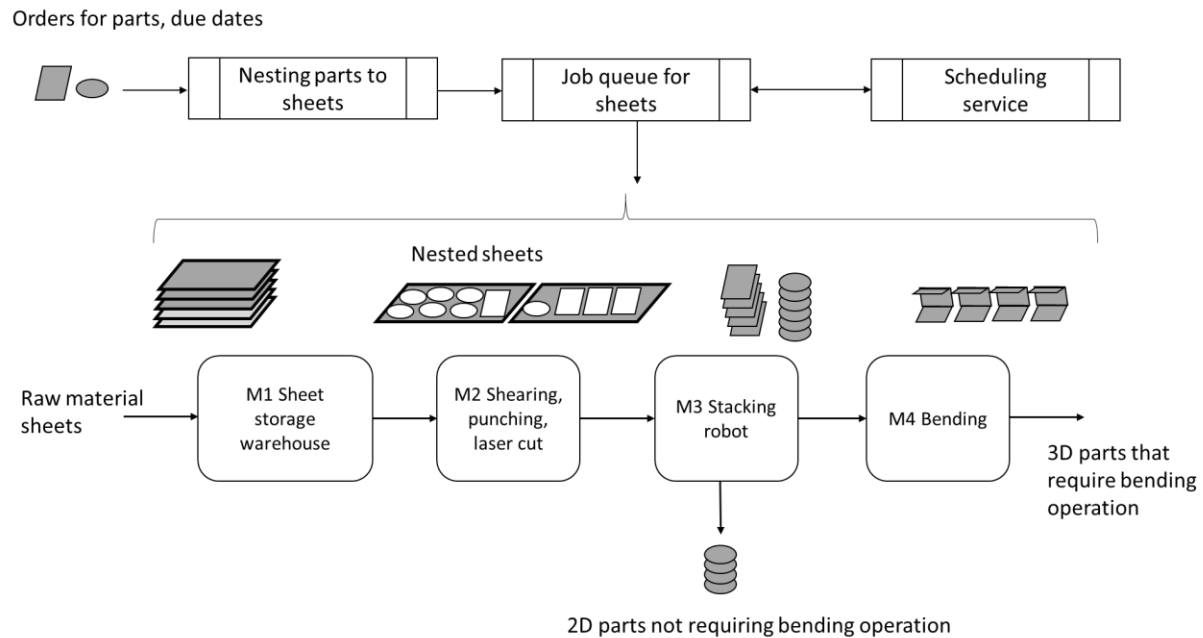
## **2. Research Background**

The cloud based Scheduling as a Service system has been implemented in sheet metal manufacturing environment. The background of the development has been to develop service solution for smart connected machinery. Additionally, the system has aimed to provide good value adding solution for manufacturing companies operating sheet metal lines, very often with limited use of advanced planning and sometimes just by using FCFS principle.

### **2.1 Sheet metal manufacturing context**

The sheet metal industry has focused on the automation of cutting, punching, forming, shearing and nesting processes for sheet metal parts (Wang & Bourne, 1997; Verlinden et al., 2007). Sheet

metal manufacturing line consists typically of several automated machines connected together, such as automated warehousing management system (WMS), laser cutting machines; punching-shearing machines, conveyors, robots and bending machines. Figure 1 demonstrates a flow shop layout of this type of combination. From scheduling point of view, the layout of the line is flowshop or flexible flowshop. Raw material sheets are stored at warehouse system, nested sheets are processed at punching-shearing and laser cut machines. Then stacking robot removes parts, and remnants. Those parts that are ready will be taken out from the line. The rest will follow toward bending machines which are forming the 3D shape to parts.



**Figure. 1.** Flow shop layout consisting of multiple WMSs (every WMS is built up with different sets of machines).

Sheet metal manufacturing has special features related to production planning and control:

- (1) Integration with design (Wei et al., 2013).
- (2) Nesting problem of the part geometrical shapes in the steel sheets (Nehal et al., 2008)
- (3) Process selection problems (Wang & Xie, 2005)

- (4) Tool change and setup problem with wearing parts (Giannakakis & Vosniakos, 2008)
- (5) Flow shop scheduling problem (Akturka et al., 2007)

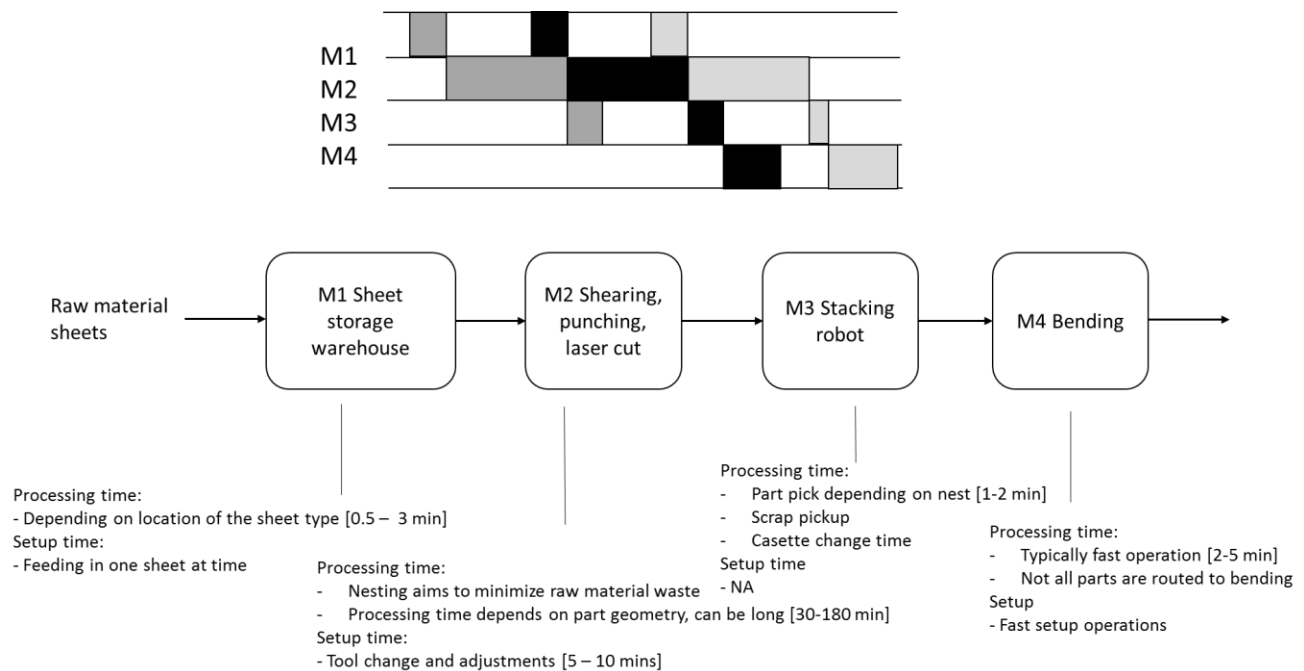
Several quantitative analytical approaches have been presented to solve these problems related to production planning and control. From production planning and control point of view, the proposed system needs to be able to handle typical situations in the environment. Based on requirements analysis, a typical sheet metal manufacturing line has the following characteristics:

- Line consisting of 5-6 machines in a flow shop type of layout. Some machines may be not used in all routings, however, the sequence of operations remain the same.
- Typically, a set of from 20 up to 200 jobs in the production queue are available for scheduling, due dates are important
- Setup time calculations are dynamic for each machine, depending on tool selection, tool alignment
- Material changes in terms of type and thickness
- Some machines may require labour operation and delayed actual operation from planned may cause delays for the entire line
- Maximum buffer size between the machines may present a constraint
- Nesting problem in sheet metal utilization is affecting the scheduling and batch sizes as parts are nested based on waste minimization only

Production planning and control system in this environment needs to be able to combine the various optimization needs for different decisions. Different planning issues and numerous problem-specific constraints need to be taking into account. Most manufacturing companies use manufacturing execution system (MES) or an equivalent system for production planning and control. Managing multi-site execution is challenging (Helo et al., 2014).

For sheet metal production job scheduling is typically adjusting the production sequence of nests. Each part is assigned on nesting optimization to a nest. Material related setup time can be avoided

by combining orders with same sheet types. Tool changes may be reduced by combining orders with similar tooling requirements. The challenge is that these requirements are conflicting. Sequence decided in the first phases defines the rest of the operations as space for stacking parts is limited. Figure 2 illustrates various conflicting machine level optimization tasks. On the high production line level, the objective is to minimize total makespan subject to due dates. Warehousing systems are quite fast and by using predefined scheduling, work can be conducted while laser cut and punch-shear operations are still taking place. For 2D operations, the nesting and tool routing defines actual time to complete a nested sheet. Tool changes may be manual or automatic and require certain amount of time in all cases. Cutting is driven by nesting as key parameter is to minimize total material cost as function of sheet utilization, material cost, remnant cost and scrap cost. Stacking robots operate in small spaces and can get stuck in case pallets are not handled responsively by the operator. Bending operations are typically fast operations and setup times are short compared to other machines in the line.



**Figure. 2.** Conflicting machine objectives for scheduling.

Objectives and operating principles can vary between production lines. High volume lines are often connected to other manufacturing phases such as painting and assembly. These lines are driven by using high capacity utilisation objective. Lower capacity utilization is typical for contract manufacturers and short due dates and rush orders present a priority.

## **2.2 Scheduling as a service in cloud manufacturing**

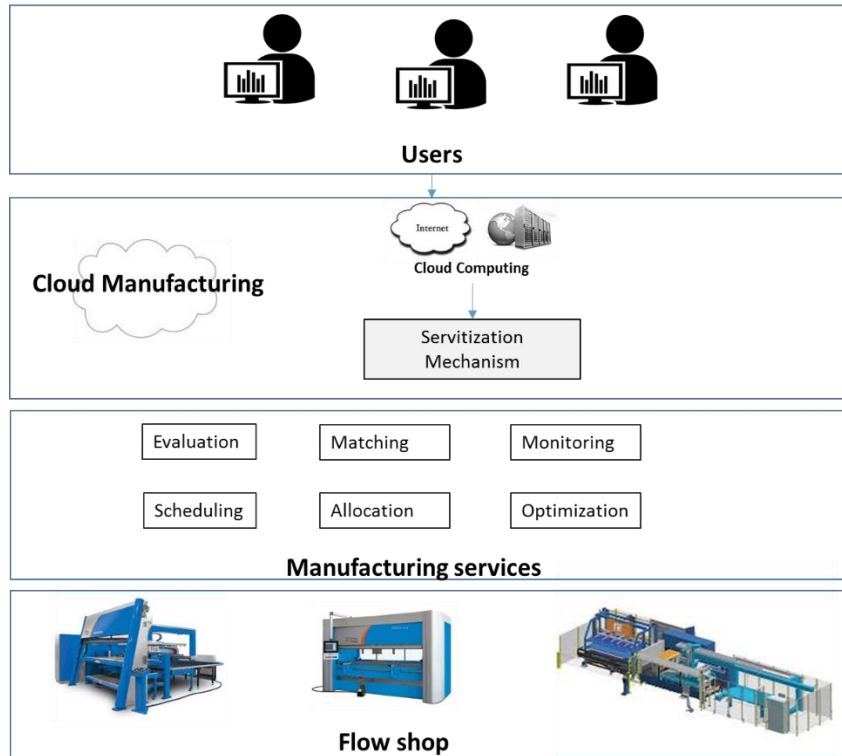
Cloud manufacturing (CM) has gradually become a popular research topic in current continuously evolving manufacturing industry. It is a new business formation to support distributed manufacturers to collaborate in the manufacturing operations, and share their manufacturing resources/capabilities. The benefits of CM are cost-efficiency, flexibility and scalability (Wang, 2013).

There are two types of interpretation of CM: smart manufacturing with cloud computing and manufacturing version of cloud (Xu, 2012). The perspective of smart manufacturing is to utilize the benefits of cloud adoption, and implement various enterprise systems on cloud based platform. On the other hand, manufacturing version emphasizes the networked relationship among different manufacturers. In this particular research, the former concept is adopted. However, these two aspects would illuminate each other. Scalable collaboration based on business requirements needs the adoption of cloud technologies (Xu, 2012).

The main purpose and interest of CM is to create a collaborative working environment (Wang, 2013). With the relation to cloud, the principle of CM is manufacturing as a service (Tao et al., 2011; Wu and Yang, 2010). All the heterogeneous and distributed manufacturing resources/capabilities are integrated in the CM's virtual resource pool centrally, and then provided as different manufacturing serveries to enable the cooperation of enterprise (Wu & Yang, 2010).

In this research, the CM and its manufacturing service integration framework is illustrated in Figure 3. Different users connect to the CM by network and they can request manufacturing services through CM. All the manufacturing services, such as evaluation, scheduling, monitoring, etc. are managed by the servitization mechanism.





**Figure 3.** Cloud manufacturing and the manufacturing service model

Planning and scheduling, such as production planning, and resources scheduling, is the core parts for implementing CM in the distributed manufacturing environment (Tao et al., 2011). Current research in CM area is mainly focused on CM's concept, business opportunities, and implementations. However, there is not much research attention on solution of scheduling as a service in CM domain. It is important to design new applications to make the planning and scheduling in CM possible. To solve the manufacturing production scheduling problem, an improved intelligent algorithm should be proposed.

For small and medium size enterprises (SMEs), adopting with CM and using manufacturing services through CM are beneficial because they can dynamically establish collaborative relationship with other partners to accomplish particular business requirements while main their competitive advantage by relying on their own core resources. Besides, SMEs can avoid large initial investment on both the manufacturing resources and the IT resources to achieve sustainable operation.

## 2.3 Genetic algorithms in Scheduling

Flow shop scheduling problem is a special case of job scheduling problem in which the jobs have to be performed in a strict order. Given a set of jobs  $J1, J2, J3...$  and a set of machines  $M1, M2, M3...$ , in flow shop each of  $J$  jobs is processed once and only once on each of  $M$  machines and for each job, its operation on machine  $M1$  must precede its operation on machine  $M2$ , and its operation on machine  $M2$  must precede its operation on machine  $M3$ , etc. (Ignall and Schrage, 1965). The performance of scheduling can be measured by one or several objectives such as flow times, makespan, or tardiness. Minimizing makespan is equivalent to minimizing idle time and minimizing waiting time on all machines. In most of the cases, job shop scheduling is a NP-hard problem (Non-deterministic Polynomial-time hard) and it is hardly to be solved with exact methods especially when it is bound to different type of constraints.

Genetic algorithm (GA) is a “stochastic optimization method” and can be “conveniently presented using the metaphor of natural evolution”: a randomly initialized group of individuals will evolve following the Darwin’s Theory of Evolution (Michalewicz et al., 1996; Gen & Cheng, 2000). Individuals with suitable characteristics will have more chance to survive and reproduce in next generation; over time the population will be spread through with these variants. In other words, in GA, individuals which have better “fitness” will have better probability to be selected in the “selection” stage and the newly generated population will be created based on these individuals and their offspring using genetic operations such as “mutation” and “crossover”; the rest will be quickly discarded. There are many variants of implementation of GA. Yet, the three most important parts about GA are chromosome encoding, fitness evaluation and selection, and GA operators (crossover and mutation):

- **Chromosome encoding:** The chromosome contains the information in which the problem is presented. This is the first starting point which needs to be solved when dealing with GA. Encoding very depends on the problem, and there are typically four different type of encoding: binary encoding, permutation encoding, value encoding, and tree encoding. In most of the cases, a binary string (a vector of 0s and 1s) is used. However, in combinatorial optimization problems, such as travelling salesman problem (TSP) or scheduling or

sequencing problems, permutation encoding (a string or numbers in sequence) is more useful.

- Fitness evaluation and selection: The fitness of each solution chromosome result is evaluated by comparing the results from fitness function  $f_i$ . For example, when minimizing total make-span, each job sequence returns and total make-span value, which may be compared. The selection process of chromosomes that will be used in the next generation is an important part of the algorithm. Fitness of each chromosome is compared and based on this, decisions are made. Four generic approaches are: (1) Roulette wheel selection, where probability is proportional to goodness of fitness function; (2) Rank selection, where probability of selection is rank of fitness among the population of chromosomes in reverse order; (3) Tournament selection is based on random selection of a set of  $s$  chromosomes, then the fittest chromosome from this group is selected to breed; (4) Elite selection method means selecting best chromosomes from the generation. The percentage can be adjusted with parameter.
- GA operators: There are many ways how to implement crossover and mutation operators in GA and this mostly depends on the nature of the problem and the type of encoding. The operations decide the evolutionary direction of population and the whole searching way of algorithm. These algorithms are suitable for solving the job scheduling problem in manufacturing and optimization problems. Crossover refers to randomly select two chromosomes as two parents and swapping them to produce two children. Mutation, referring to adding randomness in crossover operation, helps the algorithm not to converge too soon.

GA represents a good solution possibility when dealing with this type of problem because of its flexibility and the ability to transform and apply the algorithm into real problem easily. Optimization objective may be changed and modified for different production lines and situations. There are various types of heuristics and metaheuristics algorithms which can be used to optimize the production scheduling problem, yet evolutionary algorithm seems to be persuasive and suitable due to the fact that the learning curve of genetic algorithm is much shorter and it does not require much mathematical background about the optimization problem.

### 3. Problem formulation

Flow shop scheduling problems are considered to be NP-hard (Non-deterministic Polynomial-time hard), which, in computational complexity theory, is defined as “at least as hard as the hardest problem in NP”. Therefore, finding the best optimal solution would be impractical in this type of problem due to the fact that it would require a vast amount of time to find such a solution (Reeves, 1995). In practical production planning, it is more rational to find a local optimal solution within a reasonably small computer time. Local search is applied enable a straightforward way to determine optimal solutions after a limited computational effort. However, the solution is good but not necessarily optimal (Verlinden et al., 2007).

Scheduling task for sheet metal line is a variant from flow shop problem and it includes certain specific characteristics. The problem can be modelled in several ways depending on company or production line based requirements. The presented case is an example of possible model. In reality each manufacturer or production line can have different type of problem formulation and the generic purpose cloud GA solver can use the model needed.

The problem model can be parametrized into cloud solver by using data from the production planning table and up to date information retrieved directly from the machinery. The following approach aims to show a simple example which includes several parameters typical for the domain and could work as a template for company or production line specific scheduling principles

Assume a list of jobs  $\{J_1, J_2, \dots, J_n\}$  are having the following parameters, (1) processing cycle time and (2) tool related setup-time for each machine  $\{M_1, M_2, \dots, M_m\}$ , (3) due date for each job, (4) material type and thickness to describe the raw material type needed for each order. Figure 4 illustrates the details data of jobs for problem:

- Processing time and setup-time: the format will be x-x-x according to the number of machine in predefined flow sequence.
- Material related setup time: the amount of time needed when changing material type or thickness on the first punching and cutting machine. For example, if the first job is Steel 0.5 and the second one is Steel 0.6 or Aluminium 0.5, we would need an additional this amount of "unit of time" to change that on the first machine. Combining same material types together would reduce the setup-time. This time will be also included in the calculation of makespan.

- Due Date: the amount of "unit of time" before the job becomes invalid. This due date is calculated from the beginning when first job is scheduled on first machine. This will effect on the final makespan as well as the order of jobs and can be used as priority parameter as well. Value "0" means that no due date has been defined and any other value refers to exact minute.
- Material type: value can describe the type of sheet "stainless steel", "steel" or "aluminium".
- Material thickness: any numeric value describing the sheet thickness in mm.
- Grouping (nesting): indicates whether a job related to part is nested to another job (sheet metal). This will have major impact on the formula to calculate final make span as well as the visualization of job sequence on Gantt chart.
- Quantity: the amount of child job in the group. This parameter is used together with Grouping one to help identify how many child jobs are nested into the parent one.

In addition to above parameters, a job can also include the setup for tools configuration for each machine. Below illustrates how tools and child jobs are configured in detail for each machine:

- Tool stations: describes list of tools needed for each job in each machine. For example, a typical turret punch which is mostly used in punching machine in sheet metal manufacturing can have up to 60 slots that can be rotated to bring any tool to punching position. The list will affect the setup time if required to change any tool on certain machine.
- Detail child jobs: Each job can be divided into many child jobs. Each child jobs can have its own setup time and processing time on each machine. The accumulated setup time and processing time will be added to the parent job as shown in figure 4.

### Job J1

Job name

J1

Processing time

10-8

Edit

Setup time

2-2

Edit

Due date

0

Material type

Material thickness

0.5

Material size

Tool required

☒

Tools configuration

Child jobs configuration

Tool stations

Machine Id	Station	Tool	Actions
M1	C1A	T1.1	Delete
M2	B1A	T1.1	Delete

Add tool

### Job J1

Job name

J1

Processing time

10-8

Edit

Setup time

2-2

Edit

Due date

0

Material type

Material thickness

0.5

Material size

Tool required

☒

Tools configuration

Child jobs configuration

Detail child jobs

Machine Id	Child Job Id	Processing time	Setup time	Actions
M1	J1	10	2	Delete
M2	J1.1	5	1	Delete
M2	J1.2	3	1	Delete

Add child job

**Figure 4.** Detail job with tool configuration and detail job with child jobs configuration.

Setup mechanism is machine type dependent. For example, assuming in one production line there are punching and bending machines. For punching machine, there will be predefined tool stations setup, while for bending machine there will be simply basic setup times as illustrated in figure 5.

### Machine M1

Machine name

Combi

Description

Combi machine

Tool basic setup time

5

Tool change setup time

2

Predefined Tool Stations

Station	Tool	Actions
C1A	T1.1	Delete
C1B	T1.2	Delete
C2	T2.1	Delete

### Machine M2

Machine name

Bender

Description

Bender machine

Tool basic setup time

3

Tool change setup time

1

Predefined Tool Stations

Station	Tool	Actions
---------	------	---------

**Figure 5.** Setup comparison between punching and bending machines (tool configuration).

In addition to tool configuration, setup for punching and bending machine will be also much different when the output of the first machine includes many parts and each individual one will be the input for the second one. For instance, figure 6 illustrates how such a hierarchy can be setup for machine M1 and M2. Job J1.1 and J1.2 are the output of job J1 on first machine and they are input to second machine individually (they will also have separated processing time and setup time).

Detail child jobs				
Machine Id	Child Job Id	Processing time	Setup time	Actions
M1	J1	10	2	Delete
M2	J1.1	5	1	Delete
M2	J1.2	3	1	Delete
<div>Add child job</div>				

**Figure 6.** Setup comparison between punching and bending machines (child jobs configuration)

Given a job permutation  $\{J_1, J_2, \dots, J_n\}$ , a set of machine  $\{M_1, M_2, \dots, M_m\}$  and processing times  $p(i, j)$ , set up times  $s(i, j)$  for job  $i$  on machine  $j$ , we calculate the completion times  $C(J_i, j)$  - of job  $J_i$  on machine  $j$  - as follows:

$$C(J_1, 1) = p(J_1, 1) + s(J_1, 1) \quad (1)$$

$$C(J_i, 1) = C(J_{i-1}, 1) + p(J_i, 1) + s(J_i, 1) \quad (2)$$

$$C(J_1, j) = C(J_1, j-1) + p(J_1, j) + s(J_1, j) \quad (3)$$

$$C(J_i, j) = \max\{C(J_{i-1}, j), C(J_i, j-1)\} + p(J_i, j) + s(J_i, j) \quad (4)$$

for  $i=2, \dots, n$ ;  $j=2, \dots, m$

$$C_{\max} = C(J_n, m) \quad (5)$$

Where the makespan  $C_{\max}$  is the time at which the last job  $J_n$  is completed on machine  $M_m$ .

The formula for setup time  $s(i, j)$  is defined as below:

$$s(J_i, j) = (\text{basic setup time of job } J_i \text{ on machine } j) + (\text{extra setup time of job } J_i \text{ on machine } j) \quad (6)$$

where (extra setup time) is dynamic and calculated based on set of problem's constraints (e.g. material type, material size, material thickness, and tools change)

If  $C(J_i, m)$  (the complete time of job  $J_i$  on last machine) is exceeds the “Due date” of Job  $J_i$ , the order will become invalid and a penalty is put on  $C_{\max}$  (the makespan is assigned to a very large double value so that its fitness becomes very small and is likely not possible to be selected in next generation)

When a job is nested into another one or when the output of first machine is the input of second machine separately, the final makespan is calculated using the formula for nesting:

Given a job permutation  $\{J_1, J_2, \dots, J_n\}$ , a set of machine  $\{M_1, M_2, \dots, M_m\}$  and processing times  $p(i, j)$ , set up times  $s(i, j)$ , a set of child job  $\{J_{i,1}, J_{i,2}, \dots, J_{i,k}\}$  for job  $i$  on machine  $j$ , we calculate the completion times  $C(J_i, j)$  - of job  $J_i$  on machine  $j$  - as follows:

$$C(J_1, 1) = p(J_1, 1) + s(J_1, 1) \quad (7)$$

$$C(J_i, 1) = C(J_{i-1}, 1) + p(J_i, 1) + s(J_i, 1) \quad (8)$$

$$C(J_1, j) = \sum \{C(J_{1,h}, j-1) + p(J_{1,h}, j) + s(J_{1,h}, j)\} \text{ where } h=1\dots k \quad (9)$$

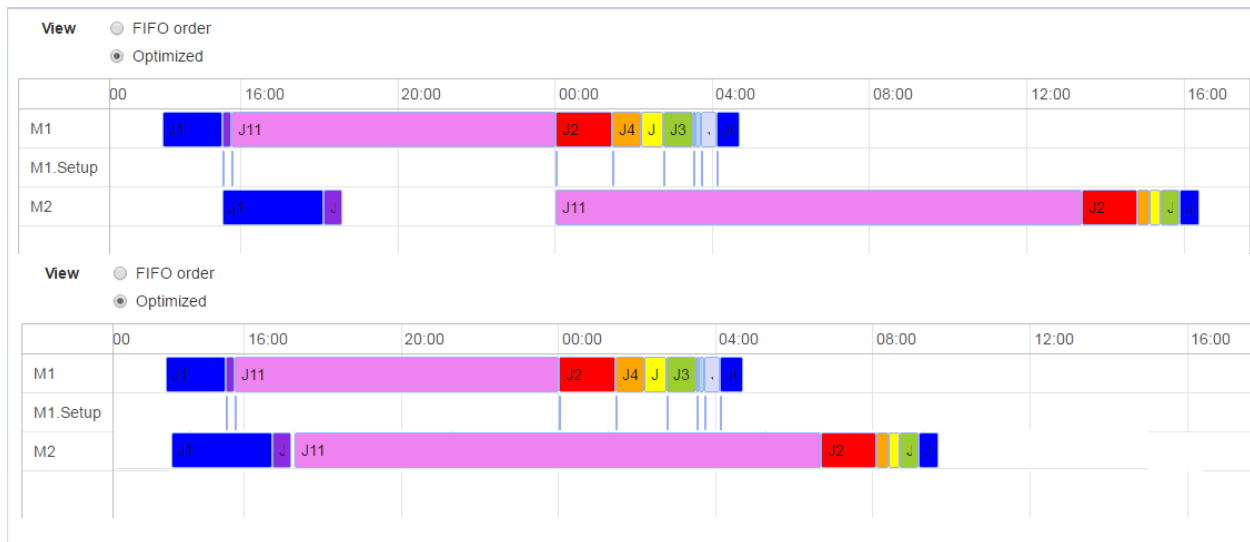
$$C(J_i, j) = \max\{C(J_{i-1}, j), C(J_i, j-1)\} + \sum \{p(J_{i,h}, j) + s(J_{i,h}, j)\} \text{ where } h=1\dots k \quad (4)$$

for  $i=2, \dots, n$ ;  $j=2, \dots, m$

$$C_{\max} = C(J_n, m) \quad (10)$$

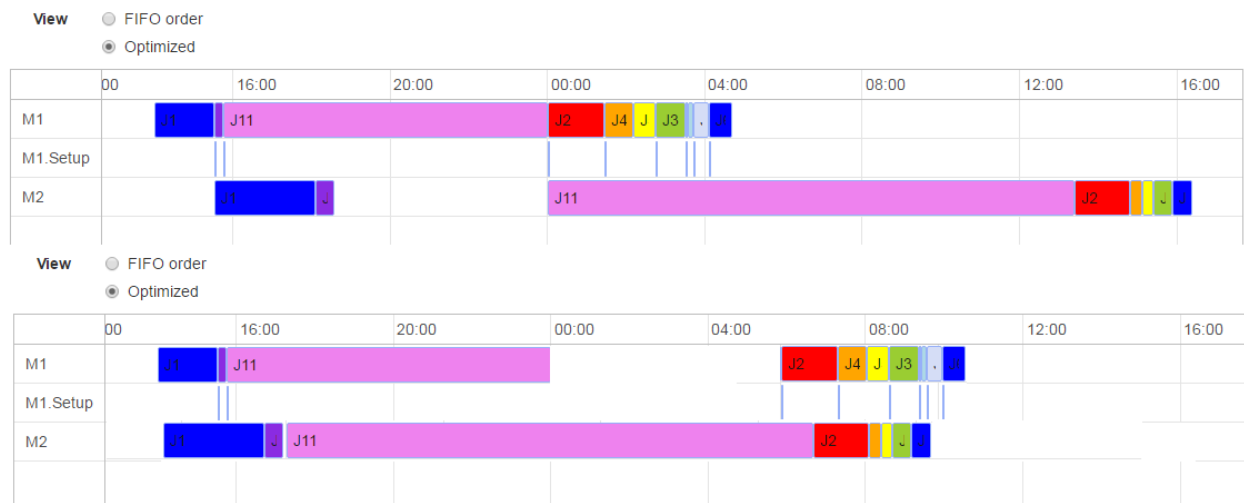


There is also distinction when visualizing the sequence of jobs and nested ones. For example, the makespan is shorten or the idling time between jobs is shorten because it has shorter processing time in grouping/nesting case (figure 7). This case could be related to with the assumption that the capacity of stacking robot (the robot machine between the first and second one in case a job is nested which acts as a small storage) is indefinitely or very large.



**Figure 7.** Distinction between normal single level job and a nested one (the latter chart)

This is an example of layout specific time calculation. In case of limited buffering capacity on this stacking robot, the situation would be slightly different as seen in figure 8 latter chart.



**Figure 8.** Distinction between normal case and nesting one with limited stacking capacity

Heuristic techniques could be used to solve some versions of the scheduling problem, but challenges are great when multiple interdependent setups are introduced as well as complicated rules for maximum line storage capacity, line blocking or robot behavior is introduced. For this reason, a metaheuristic approach is very suitable and flexible for multi-tenant cloud based solvers.

## 4. Solution approach

Each production system can have an own objective function and constraints. The scheduling task is sent by using Service Oriented Architecture (SOA) for solver located in the cloud and serving several lines on demand. The structure of GA allows separating the calculation into several parallel processors to boost the calculation speed and divide the computing capacity between different clients.

### 4.1 Computation

There are many factors which can be used to measure the performance of job shop sequencing problem. Nevertheless, in this particular flow shop scheduling problem, the objective is to find the permutation of jobs which have the minimum makespan (Reeves, 1995). Applying the traditional GA with binary encoding into sequencing problems is obviously practically difficult. Hence, in this typical one, permutation encoding is used to represent the chromosome, which is the sequence of jobs. In order to do illustrate this, we denote  $J_i$  as the unique identification for job  $i$  and in case the amount of jobs is 6, a valid chromosome can be presented as following:

*Chromosome A* =  $\{J_1, J_5, J_6, J_2, J_3, J_4\} \Rightarrow \{0, 4, 5, 1, 2, 3\}$ , where  $i=0, \dots, n$  is the index of job in the job's list

*Chromosome B* =  $\{J_2, J_3, J_6, J_1, J_4, J_5\} \Rightarrow \{1, 2, 5, 0, 3, 4\}$

The traditional GA operators cannot be applied directly into this type of problem because of its legitimate representation. For instance, the following crossover would produce “illegitimate” offspring:  $\{0, 4, 5, 0, 3, 4\}$  and  $\{1, 2, 5, 1, 2, 3\}$  - the problem would be that here jobs  $J_1$  and  $J_5$  appear twice in the first offspring and not all jobs are visible in the chromosome.

Different representations are possible, but in this case it seems easier to modify the idea of crossover and keep the permutation representation (see Reeves, 1995). For example, for one-point crossover, we firstly select the crossover point randomly. The offspring will be filled up with the index string from beginning of chromosome to the crossover point from one parent, and the rest will be filled up by taking each “legitimate” gene from the second parent, for example:

*Chromosome A*            *X*            *Chromosome B*

*Offspring 1* = {0, 4, 5, 1, 2, 3}

*Offspring 2* = {1, 2, 5, 0, 4, 3}

Next, mutation will be applied on the offspring with a small probability. The purpose of mutation is to reduce the rate of convergence. However, a modified version of mutation operator will be used instead of the traditional one, in which the positions of two random jobs will be exchanged in the chromosome:

*Offspring 1 Mutation* = {0, 4, 2, 1, 5, 3}

*Offspring 2 No mutation*

The algorithm is continued with new generated population after going through the evaluation and selection phase again if the end condition is not satisfied yet. Otherwise, the best result (chromosome) in current population will be returned and the algorithm will stop. In case of production scheduling, the best results and the winning job sequence vector, can be returned to user to see how solutions improve by time.

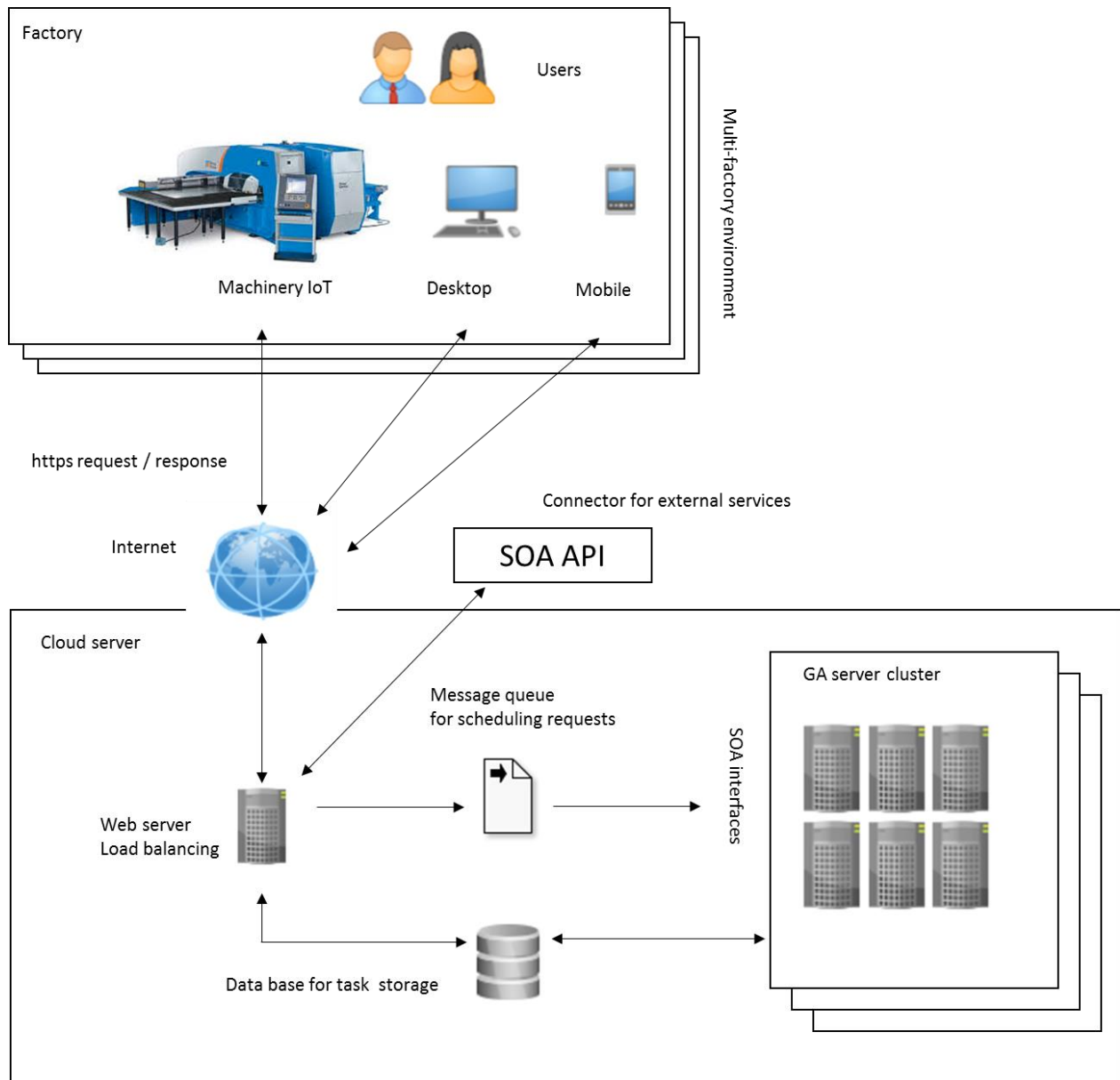
## **4.2 Technology architecture for cloud manufacturing**

Cloud computing presents a centralized process to handle the distributed data of cloud manufacturing. By using service oriented architecture in software, several clients including production planners, factory managers but also smart machinery using IoT connectors to request new solutions for changes in shop floor. Cloud manufacturing principles offer real-time

availability of the production and material related information for all the participants. It has the potential enables new possibilities to offer production scheduling as a service and competitive advantage. To solve the problems mentioned above, cloud based system need to be designed to fulfil the requirements. This cloud architecture provides the following features:

- (1) A standardized interface for exchanging production order and event information – machines updating actual processing and setup times, production planning updating the planned and released schedules;
- (2) Communication with customers and supplier about the status notifications and queries by using SOA and REST API (REpresentational State Transfer Application Programming Interface);
- (3) Centralized scheduling service for production planning and control serving several factories, production lines and machinery;
- (4) Local graphical user interface for the production line manager

Capacity, labour, material, tools and other resources can be scheduled as an input for the system. When the scheduling is requested as a service, the GA can be executed in the grid of cloud servers. Then based on demand data, each part of the production line may request schedule information or feedback any deviation from the planned processing times as output of the cloud manufacturing. The output is generated in real-time based on manufacturing situation. This is the advantage by using cloud-based technology architecture which is totally different comparing with traditional operation process.

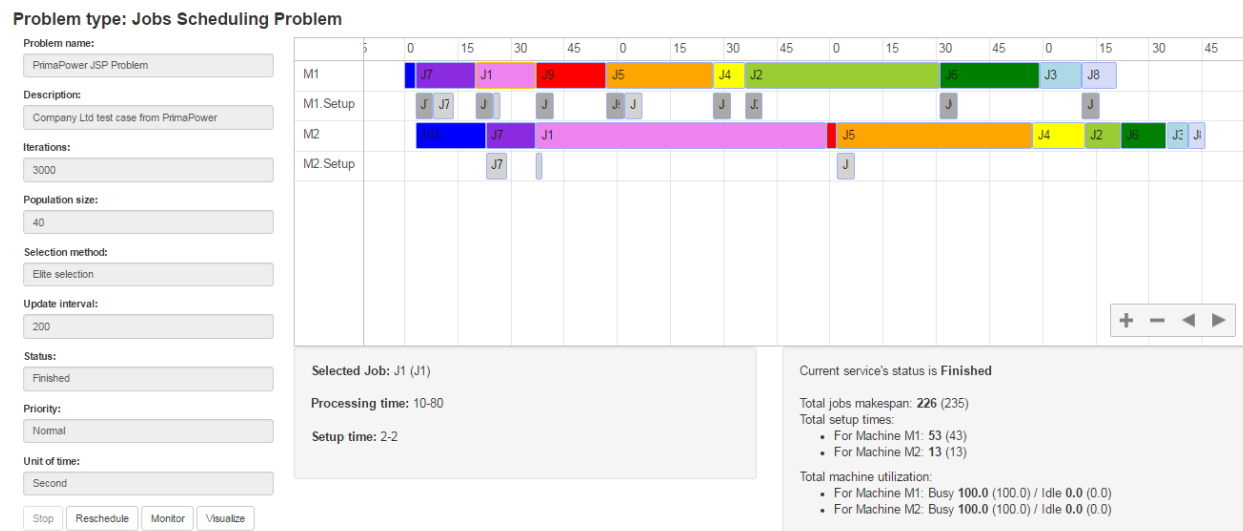


**Figure 9.** Cloud architecture for production scheduling as a service.

Figure 9 shows the cloud architecture for production scheduling as a service. Client-server architecture allows separation of user interfaces and the computation engine and the databases. A multi-tenant server software can balance computing resource load and adjust needed calculation power on demand from the cloud resources. User interfaces can be developed on local machine level or production line level. Production planners can send queries to the centralized server by

using asynchronous communication. The information required to run each machine is always stored at the local level. From architecture point the cloud system provides a centralized service for the machines of the line and transparency/visibility for system wide effects. In case of Internet communication problems, the actual production should not be endangered.

The prototype system was implemented by using Microsoft Visual Studio development tools and C# programming language. The server system was installed into Microsoft Azure private cloud system. Each user of a company can customize their system based on their manufacturing working environment. A web application with Gantt chart has been implemented to illustrate the different solutions visually, as shown in Figure 10. Gantt chart as a user interface is used to visualize the job orders, in which user can see the gap or waiting time between jobs. In the perspective of engineering, this waiting time needs to be minimized in order to minimize the makespan of problem. The algorithm will stop when user triggers “stop” button or when the current iteration reaches the pre-defined level. This interface can be visualized on different devices depends on users’ preferences, such as web browser.



**Figure 10.** User interface - local Gantt chart visualizing the proposed schedule.

Most of the real world industrial problems have to find the global optimum based on the different layers of the operations. Because in separated layers local optimized of the same variable could be

found, the real challenge is to find global optimum in multi-objective optimization problem. Optimization of production is a complex task with multiple simultaneous objectives. Each of the production line maintains information of the released production orders. As a service the cloud system can propose alternative schedules. The objective function can be a make span minimization problem, taking into account parameters such as: due date of orders; duration and possible setups for each machines; tool change including wearing; setup-times related to material, manual tool changes and operator; priorities and blocking situations.

### **4.3 Implementing genetic algorithm on cloud**

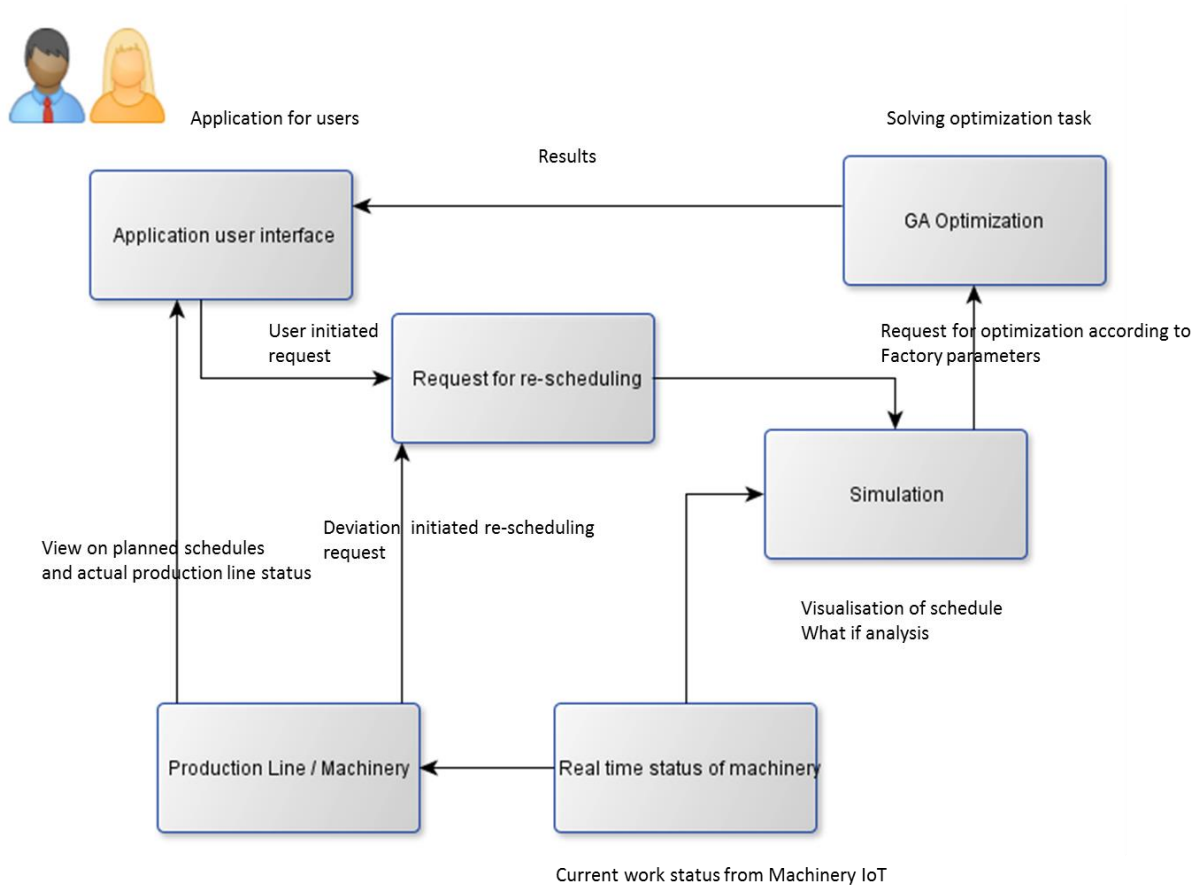
A genetic algorithm solves the scheduling problems given by the production lines. In case of blockage of production or maintenance, the production can be rescheduled or alternative scenarios can be developed. Centralized cloud based system allows separation of optimization problem and the solution generation part. Different production lines may use different principles. For example – make-to-stock type of production may put higher priority to sheet utilization and less emphasis on lead-time performance. In case of sub-contractor operating make-to-order production, the objective should be weighed the other way around. A good flexible feature of GA is that the fitness function may be tailored for different needs and the solving part can remain similar. The domains of problem formulation and problem solution are not linked heavily which allows flexibility.

In cloud manufacturing, all the production lines are constantly monitored at runtime, so the scheduling should be dynamically coordinate with the real situation of production lines. Figure 11 shows the major elements of scheduling service:

- (1) The users of cloud manufacturing can select the production lines they want to schedule and simulate. Then the user requests will send to the cloud manufacturing to facilitate the optimization and simulation task.
- (2) The information regarding the real-time status of particular production lines are streamed back to the cloud manufacturing used as another essential input of simulation task.
- (3) The desired algorithms are input into the scheduling. When the input data are received by the simulation task, then new optimization results can be created.

- (4) Once the output is completed, the optimized results will send back to users. When the users accept the plan, it can be initiated by the particular production line.
- (5) Smart IoT machinery can request new schedule based on event triggered in the production line.

Additionally, as planned and actual processing and setup times are updated close to real-time in centralized cloud, re-scheduling can be evoked automatically. The role of the production planner is to approve the suggested changes according to new situation.



**Figure 11.** The main elements of scheduling service in cloud manufacturing.



## 5. Results

In order to demonstrate the validity and reliability of the implemented cloud based GA, we have compared the value of make span which is calculated using GA on a single processor and the one which is calculated using the normal “*for...loop*” operation in simple cases.

For example, if the number of jobs is 8, and the number of machines is 3, we would need to calculate maximum  $8! = 40320$  permutations of jobs in order to find the best optimized value. This amount of calculation is quite small and it can be completed within few seconds in a standard computer. Using the same set of machines, jobs and processing times, the best optimal value of make span can be obtained by both solutions and in nearly most of the cases in Genetic Algorithm, the best value can be reached after 100 iterations. To analyse the performance of the system a randomized dataset was developed for 100, 200 and 300 jobs, which refer to typical problem size. Table 1 shows the performance of the implemented GA measured by the makespan and the execution time in “more likely NP-hard” cases. The make span is recorded by executing the same GA setting three times continuously and the best value is returned. It is also noted that the more jobs we have, the more time we need to complete one single iteration because more calculation needs to be executed for each iteration proportionally.

**Table 1.** Performance of GA measured by makespan and execution time (Intel Core i7-3630QM CPU 2.4 GHz).

No. of jobs / No. of iterations	I = 1000	I = 5000	I = 10000
N = 100	2910	2901	2901
	3.13s	14.48s	29.18s
N = 200	5550	5537	5537
	5.97s	28.06s	54.42s
N = 300	7822	7819	7819
	10.83s	61.06s	109.98s

As it can be seen from the results, a ‘good’ value of makespan can be achieved quite quickly in GA and the value does not change much comparing to the number of iterations need to be executed. In other words, GA is extremely efficient when dealing with complex problem, in which it is more practical to find a ‘good enough’ solution within a reasonable small computer time. The objective function can be modified and solving part of the GA system can remain same, which brings flexibility needed for solving multi-objective and constraint-driven problems. The more complicated the problem is, the more efficient the GA can be comparing to other heuristic search methods. However, there is no guarantee of optimality of solution in GA. Table 2 shows experimentation of 10, 30 and 50 jobs and number of iterations for 1000, 5000 and 10000. For small problem, the optimality can be reached fast, for bigger problems the reaching better solutions closer to optimality takes more iterations.

**Table 2.** Best solutions and deviation from optimal value in known flow shop scheduling problem (Intel Core i7-3630QM CPU 2.4 GHz).

No. of jobs / No. of iterations	I = 1000	I = 5000	I = 10000
N = 10 (optimal makespan=263)	266 (+3)	263 (0)	263 (0)
N = 30 (optimal makespan=4743)	4851 (+108)	4815 (+72)	4780 (+37)
N = 50 (optimal makespan=21233)	21804 (+571)	21569 (+336)	21410 (+177)

## 6. Conclusions

In this paper we have presented a concept of genetic algorithm based scheduling system offered as part of cloud infrastructure for sheet metal manufacturing. The system has been developed and piloted at a sheet metal processing line environment. The advantage of this proposed scheduling system is that cloud manufacturing is capable of supporting the close to real-time analysis and planning with dynamic requirements and inputs for multiple factories and connecting the actual processing times from machinery.

The innovation of this study has been to develop the cloud-based scheduling as a service architecture which can serve several machine lines, factories and companies for continuous rescheduling requests. The solution can be implemented by machine builders as a service product. The developed system is a pilot in its current form. It is able to offer scheduling jobs from the cloud on request for multiple production lines and provide information between entities in the supply chains. Traditionally, this type of operations are carried out typically by using separate software packages or integrated factory level manufacturing execution systems. Based on the piloting the cloud based system enables several possibilities compared to traditional production scheduling systems:

- (1) Production scheduling can be done continuously, not daily as the data from machines can be fed back to cloud scheduler
- (2) Actual processing times may be updated system real-time from each machine and database on job - machine or work phase specific error data could be input directly to PPC system
- (3) Alternative schedules can be adjusted based on feedback and situations by sending new queries to scheduling server.
- (4) Distributed user interfaces can show current status to all participants along the supply chain.

The presented cloud approach has both business and technical advantages. From the business perspective, the implementing scheduling as a service on cloud manufacturing can aggregate the optimization problems and provide integrated solution to manage distributed production lines. Production line vendors can offer scheduling as a service to support high utilization of lines and

provide feedback of actual use of the lines. From the technical perspective, the cloud-based architecture can deal with multi-objectives optimization problems in a much faster manner and absorb the problem of limited computing resources to accomplish the optimization task. Cloud based OR systems can provide significant help when embedded close to machinery as “artificial intelligence” services.

Cloud manufacturing concept is still very much evolving. According to Chandrasekaran et al. (2013) many issues are confusing since practical applications are few. The present work is a step toward demonstrating the applicability of cloud computing concepts to optimization as a service and centralized production scheduling services. Production scheduling as a computational intensive service could benefit from centralized service. High calculation power is needed occasionally and peaks may occur at different times. New approaches on high performance GPU-processors cloud and new advances in GPU based parallel GA optimization can show good potential for future research in industrial applications of production scheduling.

## Acknowledgements

Financial support from Support from the Finnish Metals and Engineering Competence Cluster (FIMECC) S-Step4 program is gratefully acknowledged.

## References

- Akturk, M. S., Ghosh, J. B., & Kayan, R. K. (2007). Scheduling with tool changes to minimize total completion time under controllable machining conditions. *Computers & operations research*, 34(7), 2130-2146.
- Chandrasekaran, M., Muralidhar, M., & Dixit, U. S. (2013). Online optimization of multipass machining based on cloud computing. *The International Journal of Advanced Manufacturing Technology*, 1-12.
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 182–197. doi:10.1109/4235.996017
- Fujii, S., Kaihara, T., & Morita, H. (2000). A distributed virtual factory in agile manufacturing environment. *International Journal of Production Research*, 38(17), 4113-4128.

- Ford, S. J., Rauschecker, U., & Athanassopoulou, N. A. (2012). System-of-system Approaches and Challenges for Multi-Site Manufacturing. *Proceedings of the 2012 7th International Conference on System of Systems Engineering*, Genoa, Italy-16-19 July 2012
- Gen, M., Cheng, R. (2000). *Genetic Algorithms and Engineering Optimization*. John Wiley & Sons.
- Giannakakis T. And Vosniakos G. C. (2008). Sheet metal cutting and piercing operations planning and tools configuration by an expert system, *International Journal of Advance Manufacturing Technology*, 36, pp.658-670.
- Hao, Y., & Helo, P. (2014). A new paradigm of manufacturing management: Cloud manufacturing. *Proceedings of The 18th Pacific Asia Conference on Information Systems (PACIS 2014)*, Chengdu.
- Helo, P., Suorsa, M., Hao, Y., & Anussornnitisarn, P. (2014). Toward a cloud-based manufacturing execution system for distributed manufacturing. *Computers in Industry*, 65(4), 646-656.
- Huang, B., Li, C., Yin, C., & Zhao, X. (2013). Cloud manufacturing service platform for small-and medium-sized enterprises. *The International Journal of Advanced Manufacturing Technology*, 1-12.
- Ignall, E., Schrage, L. (1965). Application of the Branch and Bound Technique to Some Flow-Shop Scheduling Problems. *Operations Research* 13, 400–412. doi:10.1287/opre.13.3.400
- Luo, Y. L., Zhang, L., He, D. J., Tao, F., Ren, L. & Tao, F. (2011). Study on multi-view model for Cloud manufacturing. *Advanced Materials Research*, 201, 685-688.
- Michalewicz, Z., Dasgupta, D., Le Riche, R.G., & Schoenauer, M. (1996). Evolutionary algorithms for constrained engineering problems. *Computers & Industrial Engineering* 30, 851–870. doi:10.1016/0360-8352(96)00037-X
- Nehal I. Joshi, Changa Dist Anand, Ashish M. Desai (2008), Nesting Algorithms for Placement of Regular & Irregular Shaped Parts, *International Journal of Engineering Research & Technology*, pp. 1-6.
- Putnik, G. (2012). Advanced manufacturing systems and enterprises: Cloud and ubiquitous manufacturing and an architecture. *Journal of Applied Engineering Science*, 10(3), 127-134.
- Pospíchal, P., & Jaros, J. (2009). Gpu-based acceleration of the genetic algorithm. *GECCO competition*.

- Reeves, C.R. (1995). A genetic algorithm for flowshop sequencing. *Computers & Operations Research*, Genetic Algorithms 22, 5–13. doi:10.1016/0305-0548(93)E0014-K
- Tao, F; Zhang L.; Venkatesh, V.C.; Luo, Y.L. & Cheng, Y. (2011). Cloud manufacturing: a computing and service-oriented manufacturing model. Proceedings of the Institution of Mechanical Engineers, Part B, *Journal of Engineering Manufacture*. doi:10.1177/0954405411405575.
- Verlinden, B., Cattrysse, D., Crauwels, H., Duflou, J., & Van, D. (2007). Sequencing and scheduling in the sheet metal shop. *Multiprocessor Scheduling*, 345.
- Wang, L. (2013). Machine availability monitoring and machining process planning towards Cloud manufacturing. *CIRP Journal of Manufacturing Science and Technology*, 6(4), 263-273.
- Wang G. G. & Xie, S. Q. (2005). Optimal process planning for a combined punch-and-laser cutting machine using ant colony optimization, *International Journal of Production Research*, pp. 2195–2216.
- Wang, C. H., & Bourne, D. A. (1997). Design and manufacturing of sheet-metal parts: using features to aid process planning and resolve manufacturability problems. *Robotics and Computer-Integrated Manufacturing*, 13(3), 281-294.
- Wei, Z., Feng, Y., Wang, X., Sun, Z., & Sun, Y. (2013). Design Thinking and Cloud Manufacturing: A Study of Cloud Model Sharing Platform Based on Separated Data Log. *Advances in Mechanical Engineering*, 2013.
- Wu, L., & Yang, C. (2010). A solution of manufacturing resources sharing in cloud computing environment. In *Cooperative Design, Visualization, and Engineering* (pp. 247-252). Springer Berlin Heidelberg.
- Wu, D., Greer, M.J., Rosen, D.W., & Schaefer, D. (2013). Cloud Manufacturing: Strategic Vision and State-of-the-Art. *Journal of Manufacturing Systems (JMSY)*, 32(4): 564-579. DOI: 10.1016/j.jmsy.2013.04.008.
- Wu, D., Rosen, D.W., Wang, L., & Schaefer, D. (2015). Cloud-Based Design and Manufacturing: A New Paradigm in Digital Manufacturing and Design Innovation. *Computer-Aided Design*, 59, 1-14. <http://dx.doi.org/10.1016/j.cad.2014.07.006>

Xu, X (2012). From cloud computing to cloud manufacturing. *Robotics and Computer-Integrated Manufacturing*. 28(1), 75-86. doi:10.1016/j.rcim.2011.07.002

Zhang, L., Luo, Y., Tao, F., Li, B. H., Ren, L., Zhang, X., & Liu, Y. (2014). Cloud manufacturing: a new manufacturing paradigm. *Enterprise Information Systems*, 8(2), 167-187.