Proceedings of the ASME 2016 Dynamic Systems and Control Conference
DSCC2016
October 12-14, 2016, Minneapolis, Minnesota, USA

DSCC2016-9908

PRODUCTION AS A SERVICE: OPTIMIZING UTILIZATION IN MANUFACTURING SYSTEMS

Matthew Porter
Department of Mechanical Eng.
University of Michigan
Ann Arbor, Michigan 48109
Email: matthepo@umich.edu

Vikram Raghavan
Ross School of Business
University of Michigan
Ann Arbor, Michigan 48109
Email: vkrmrgvn@umich.edu

Yikai Lin
Department of Computer Science
and Engineering
University of Michigan
Ann Arbor, Michigan 48109
Email: yklin@umich.edu

Z. Morley Mao
Department of Electrical Eng.
and Computer Science
University of Michigan
Ann Arbor, Michigan 48109
Email: zmao@umich.edu

Kira Barton
Department of Mechanical Eng.
University of Michigan
Ann Arbor, Michigan 48109
Email: bartonkl@umich.edu

Dawn Tilbury
Department of Mechanical Eng.
University of Michigan
Ann Arbor, Michigan 48109
Email: tilbury@umich.edu

ABSTRACT

While advances in technology have greatly improved the process of mass production, producing small batches or one-offs in an efficient manner has remained challenging for the manufacturing industry. Additionally, in both large and small companies, there are often available manufacturing resources that sit idle between projects. In this paper we present a Production as a Service framework for providing manufacturing options to designers of new products based on available manufacturing resources. The designed framework aims to bridge the gap between the theoretical work that has been done on Service Oriented Architectures in manufacturing, and what is required for implementation. An industrial use case is provided as an example of the framework.

1 Introduction

Increasing variability in production volume and customization have driven manufacturing companies towards the more flexible paradigm of mass customization as discussed in [1]. While there are many advantages to implementing mass customization such as increased sustainability and adaptability, there are significant challenges related to the need for large investments in infrastructure and organizational changes. Mass customization has been able to succeed on a company level by following the general rules proposed by [2], but a more generalized approach that leverages existing infrastructures and production capabilities remains an open area of interest for researchers.

One common approach to address this need includes applying the concept of Service Oriented Architecture (SOA) towards reorganizing the structure of a manufacturing enterprise. In [3], an upper level description of a layer-based structure is proposed to allow for SOA to be used in a cloud-based manufacturing format. The requirements for applying SOA to a production system are described further in [4], including several simulation and computational examples, but without the validation on an experimental platform. To implement a service-oriented architecture in a production environment, the Device Profile for Web Services (DPWS) specification was used in [5]. In [6] the necessary web infrastructure is further discussed and a proof of concept is given for a generalized network of resources. Connections were made to the Delmia Automation 3D simulation models in a system-level simulation platform, exposing several issues that must be faced when rolling out SOA into a full-scale factory. A SOA was implemented on a small production system in [7], where existing architectures such as DPWS and OPC-UA were combined.

Most of these examples focus on a single plant (or at least a single company); there has been little to no discussion of how a SOA could be expanded to include several companies. While [8] provides a high level description of how SOA might be implemented on a multi-company network, it does not discuss many of the challenges that this extension would pose. Applying SOA to a group of companies would require external interfaces that preserve proprietary details of the manufacturing system. Additionally, these examples do not include discussions of how important factors such as production costs, time, and quality can be calculated for a given production option. Existing literature does not fully explore the concept of choosing between potential production options that arise from a SOA based framework especially in a multi-company setting. There has been a great deal of work in the related field of optimal supplier sourcing such as [9] and [10]. Much of this work focuses on specific applications where the customers needs, preferences, and use case are integrated into the optimization scheme. In order for optimization to be applied to a diverse SOA setting, the criterion for optimality must first be generalized, and individual customer needs and preferences must be considered. These gaps need to be addressed before a Service Oriented Architecture can be utilized on a large, multi-company scale for enhanced manufacturing flexibility and productivity.

The remainder of this paper consists of the following. Section 2 introduces a Production as a Service framework. Section 3 presents a generic methodology for identifying an optimal production solution based on user-weighted importance of three core elements: cost, time, and product quality. A case study demonstrating the proposed framework is given in section 4. Concluding remarks are provided in section 5.

2 PaaS Framework

A Production as a Service (PaaS) Framework has been designed to be a web-based platform for customers and suppliers to interact in a mutually beneficial manner. In the context of this paper the term *customer* will be used to describe users that are requesting a specific product to be manufactured, while the term *supplier* will be used to describe manufacturers that are looking to provide production services. The PaaS Framework was designed with several goals in mind:

Flexibility and Robustness: The service should support the diverse nature of manufacturing processes.

Extensibility: The service should be able to easily evolve to support significant requirements and options.

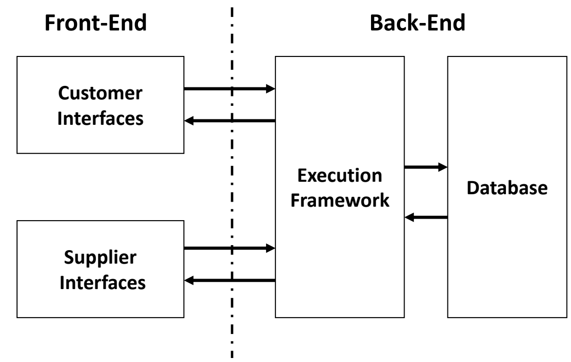

Figure 1. Overview of PaaS Framework Structure

Ease of Use: The interfaces should be useable by someone without manufacturing expertise.

Privacy and Confidentiality: Sensitive information from both the customer and supplier should be protected.

These goals are necessary for the design to be practical and their influence is discussed throughout the paper.

2.1 Front-End Structure

On a high level, the framework structure can be broken into two parts: the front-end and the back-end, as illustrated in figure 1. The front-end of the PaaS framework can be further broken down into two core segments: an interface for the customers, and an interface for the suppliers.

2.1.1 Customer Interface

From the customer perspective, there are three user interfaces that must be navigated. In the main interface, the customer can view and manage production requests that have already been submitted. A second interface allows for the generation of new production requests, while the third provides an interface for searching through the production options once a sufficient number of quotes have been submitted by suppliers and optimized based on user defined criteria.

In the second interface, new requests are generated from customer information including: (1) critical manufacturing tasks that are required to fabricate the product, (2) acceptable ranges for fabrication costs, (3) maximum fabrication times, and (4) desired part quality and quantity. The customer may choose to include supporting files such as prints or computer aided drawings (CAD) of the product, although this is not necessary at this stage of the request. Detailed CAD images will be required for final part production. Once a production request is submitted, the back-end search engine of the PaaS Framework will solicit candidate suppliers for quotes.

The customer search interface provides an overview of the manufacturing options based on available resources optimized to meet the customer criteria in terms of cost, time-line and part quality. The solutions may be from a single manufacturer, or spread across multiple manufacturing facilities. This information includes the range for cost, time and quality, along with plots of the trade-offs between each of the variables based on how the weights are chosen. This allows the customer to determine the trade-offs required to meet their preferences and to update the limits for cost, time and quality as necessary. The search process consists of an optimization strategy described in more detail in subsection 3.4. The optimization results provide sufficient detail for the customer to evaluate the different options and either select one of the provided options or refine the search further.

2.1.2 Supplier Interface

The supplier interface consists of two lists: the first list contains new requests for manufacturing quotes to which they can choose to respond; the second list contains a history of quotes that the supplier has submitted. Additionally, the interface provides a database in which the manufacturer uploads critical information such as their manufacturing resources (e.g. machining capabilities, achievable part volumes, material compatibilities, etc.) and company location. This information is used to match the supplier with new production requests that can be satisfied with their manufacturing resources.

The product request list provides the supplier with the full details of a given request including: (1) the manufacturing tasks required to produce the product, (2) the product design specifications including cost, time-line, and quality and quantity requirements, (3) the customer's contact and shipping information, and (4) any additional documents such as technical drawings submitted by the customer. Selecting a request to respond to yields another interface that allows the supplier to generate quotes. The supplier can create multiple quotes for a given request by itemizing the quote into individual steps or by quoting subsets of steps.

2.2 Back-End Structure

The PaaS back-end supports the data processing and computation aspects of the framework. Data received from the front-end, such as the supplier quotes, will be processed by the back-end and stored properly and securely for future access. Once a customer initiates a new query, the back-end retrieves data from a storage database, performs the necessary optimization, and responds with the data digest that meets consumer requirements while preserving a certain level of privacy of the data source.

Conceptually, the PaaS framework back-end is a set of software frameworks running on a cluster of computers. Building the back-end on a cluster provides the processing capability necessary for the potentially large amount of data and reliability in case of hardware failure. Each software framework runs as a module that intercommunicates through messages, and thus can be easily upgraded for extensibility. The main functionality of the back-end can be broken down into three layers, each responsible for different tasks.

2.2.1 Application Layer

The Application layer is where the programs of user-facing components are run. Some core applications include the web-server, the query translator, and the quote processor. Each functionality, implemented as a separate program, provides the perfect isolation for simpler maintenance, upgrading, debugging, and introduction of new services.

2.2.2 Control Layer

The Control layer acts as an intermediate layer between the application layer and the underlying data storage. Most of the back-end APIs are provided by this layer, such as querying for quotes, updating supplier capability, storing consumer information, etc. By isolating direct data access from the applications and exposing unified APIs, the Control layer not only enables simpler application development, but also provides extra protection of consumer and supplier data.

2.2.3 Storage Layer

The Storage layer is designed with the data format in mind. For supplier capability data, which is well structured information and may require regular updates from the supplier, an SQL like database is used. Production requests and quote information, show the pattern of appending rather than updating (ie. the information is not reused). In this case a distributed file system such as Hadoop [11] is more suitable. Therefore, the Storage layer is heterogeneous with different storage frameworks, maximizing the performance of the framework and ease of programming.

3 PaaS Functionality

The process of matching a customer with a supplier or set of suppliers can be broken down into four main steps:

1. Creation of a production request
2. Soliciting supplier quotes
3. Submission of quotes
4. Search and optimization

The first three steps are important, with general descriptions of these processes included in the manuscript. A more detailed analysis of each of these steps has been left for future work. The focus of this paper is on the development of the search and optimization methodology for the PaaS Framework.

3.1 Creating a Production Request

In addition to supplying basic customer information (e.g. shipping location, cost range, quality, quantity, etc.), the customer must describe the manufacturing processes required to fabricate their product. This knowledge is critical for determining the selection of candidate suppliers that meet these manufacturing requirements. To aid in the search engine, the production steps are standardized between the customer and supplier by requiring processes to be chosen from a list of supported processes (e.g. Milling, Turning, Stamping, etc.). This also allows the framework to prompt the customer to provide necessary information specific to a given process. For example, when selecting a milling process the customer will be prompted to provide the rough dimensions, the weight of the part, the material being machined, and the tightness of the tolerances. Additionally, the user can input further constraints such as the type of milling machine needed, although this is not a formal requirement. Together the process name and attributes are referred to as a *process definition*. Though this method does create some limitations in terms of process diversity, it allows for extensibility. Expanding the number of supported processes can be done by creating additional process definitions. Changes to the required and non-required process information can also be done by updating the process definitions.

3.2 Soliciting Quotes

Once a production request has been submitted, the database of currently available resources is queried to find suppliers with available resources that can complete any or all of the given production steps. Because the suppliers use the same standardized process definitions as those provided to the customer for creating the production request, this matching becomes quite simple. If the specifications of a given supplier's capabilities meet the needs of one or more production steps, a request for a quote is sent. It may be beneficial for a supplier to include additional information about their capabilities such as milling machine type, but once again this is not required.

3.3 Submitting quotes

After requests are sent out to candidate suppliers, time is allowed for each of the suppliers to generate quotes. Suppliers can submit quotes for any set of steps they choose and can submit as many quotes as they want. This allows suppliers to provide quotes that would only be considered for the entire job, or a portion of the production steps if desired. Each quote includes a description of the production cost, time to complete, and a quality factor ranging from 0-100%. The quality factor is based on various quality metrics attributed to the specific supplier, gathered from supplier profile information and previous interactions with customers.

3.4 Optimization

After the supplier quotes have been submitted, an optimization must be performed to generate feasible manufacturing options. In general the goal of the search is to find the optimal production request that minimizes the production cost and time, while maximizing production quality. This can be expressed as a multi-objective optimization function:

$$\min \quad (C(p), T(p), 1 - Q(p)) \tag{1}$$

subject to

$$p \in P \tag{2}$$

where $C(p), T(p)$, and $Q(p)$ represent the production cost, time and quality, respectively, as functions of the production option p. Here p is in the set of all possible production options P. The full set of production options P can be found by using supplier quotes to generate a graph structure similar those used by [9] to describe supply chains. The graph structure is created by representing quotes as nodes of the graph and connecting the nodes based on the production steps they represent. The nodes themselves are

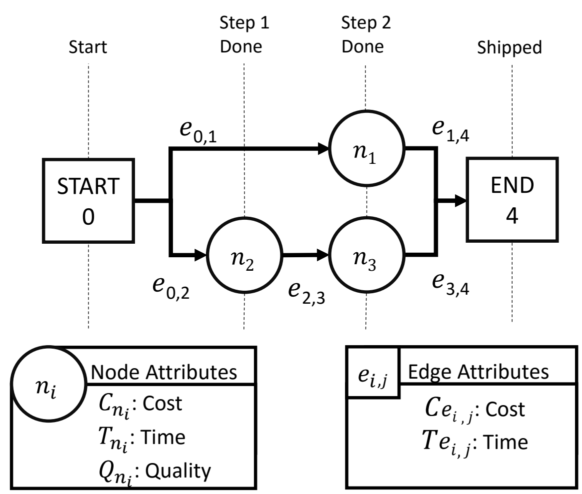

Figure 2. Simple Graph Structure Example

weighted based on the cost, time, and quality associated with the quote they represent. Edges are weighted based on the cost and time associated with shipping between suppliers. An example of this type of graph structure is shown in figure 2. This example shows a manufacturing process with two steps and three potential suppliers, one of which (n_1) has quoted the entire process while the others (n_2, n_3) have each quoted a single step. Using this graph structure, the full set of production options can then be found by completing an exhaustive search of the graph.

Allowing a production option p to be expressed as the ordered set of m nodes and edges, $\{n_1, e_{1,2}, n_2, \ldots, e_{m,m+1}\}$, corresponding to a given solution of the graph structure, the calculations for production cost, time and quality are shown in equations(3)-(5).

$$C(p) = \sum_{i=1}^{m} (C_{n_i} + C_{e_{i,i+1}}) \tag{3}$$

$$T(p) = \sum_{i=1}^{m} (T_{n_i} + T_{e_{i,i+1}}) \tag{4}$$

$$Q(p) = \sum_{i=1}^{m} \left(\frac{Q_{n_i} s_i}{\ell} \right) \tag{5}$$

In equations(3) and (4), C_{n_i} and T_{n_i} are the ith node cost and time respectively, and $C_{e_{i,i+1}}$ and $T_{e_{i,i+1}}$ are the edge cost and time. In eq. (5), Q_{n_i} is the quality associated with the ith node, s_i is the number of manufacturing steps associated with the ith node, and ℓ is the total number of steps in the manufacturing process. With this additional weighting to the individual quality values, the resulting value of $Q(p)$ becomes the average quality of the quotes in the production option with each of the individual quality values weighted by the number of production steps in the corresponding quote.

Difficulties with this type of optimization arise when the set of quotes becomes large. This makes an exhaustive search no longer feasible, and the resulting set of pareto optimal solutions too large for the customer to consider on an individual basis. Instead, the multi-objective optimization is transformed into a single objective optimization problem by normalizing and weighting each of the different objectives as shown in eq. (6).

$$\min \quad \left[\left(\frac{C(p)}{C_N} \right) + 2^{\alpha_1} \left(\frac{T(p)}{T_N} \right) + 2^{\alpha_2} \left(\frac{1-Q(p)}{1-Q_N} \right) \right] \tag{6}$$

subject to

$$\begin{aligned} &p \in P \\ &T(p) < T_m \\ &C(p) < C_m \\ &Q(p) > Q_m \end{aligned} \tag{7}$$

In eq. (6), the constants α_1 and α_2 are used to weigh the importance of time and quality relative to cost. The weights are used as exponents to allow the resulting coefficients to have much greater range, effectively allowing individual objectives to be dominated by others. T_m, C_m, and Q_m are optional constraints that can be added to the range of each of the objectives; otherwise, the full range will be used.

C_N, T_N, and Q_N are the estimated averages for each of the objectives and are used to normalize the values. The estimated averages were chosen as opposed to other values such as the maximum value in order to mitigate the effects of outliers. Also, by estimating the averages as opposed to finding the true averages the need for an exhaustive search of the graph is avoided. The constant C_N is calculated by adding together the estimated process costs for each of the steps in the manufacturing processes. These individual costs are calculated by finding the least squares solution to the over-determined system shown in eq.(8).

$$\begin{bmatrix} \gamma_{C1} \\ \vdots \\ \gamma_{Cj} \end{bmatrix} = \begin{bmatrix} A_{1,1} & \ldots & A_{1,k} \\ \vdots & \ddots & \vdots \\ A_{j,1} & \ldots & A_{j,k} \end{bmatrix} \begin{bmatrix} E_{C1} \\ \vdots \\ E_{Ck} \end{bmatrix} \tag{8}$$

$$C_N = E_{C1} + E_{C2} + \ldots + E_{Ck} \tag{9}$$

In eq. (8), $\gamma_{C1} ... \gamma_{Cj}$ are the costs associated with each of the j quotes, $E_{C1} ... E_{Ck}$ are the estimated costs associated with the k manufacturing steps, and $A_{x,y}$ is a 1 if quote x includes manufacturing step y, otherwise the entry is 0. Equation (8) assumes that there is a sufficient variety of quotes to make the A matrix of rank k. T_N can be calculated in much the same way by substituting the times associated with each quote $\gamma_{T1} ... \gamma_{Tj}$ into eq. (8) in place of the values for $\gamma_{C1} ... \gamma_{Cj}$, and then adding together the estimated time values for each of the production steps. To calculate Q_N, the weighted quality of each quote $\overline{\gamma_{Q1}} ... \overline{\gamma_{Qj}}$ expressed in eq. (10) is substituted into eq. (8) in place of $\gamma_{C1} ... \gamma_{Cj}$, and then adding together to estimate the values for each of the production steps.

$$\overline{\gamma_{Qi}} = \frac{\gamma_{Qi} \ s_i}{\ell} \tag{10}$$

In eq. (10), γ_{Qi} is the quality factor, s_i is the number of steps associated with the ith quote, and ℓ is the total number of manufacturing steps. This additional weighting is necessary to mirror the weighting used to calculate the value of $Q(p)$ in eq. (5).

Using the single objective optimization function and a simple graph search algorithm such as Dijkstra's algorithm [12], a single optimal solution can be found. Additional similar options can be found using Yen's algorithm [13] and the same optimization function. By adjusting the weights and the range constraints, the optimal solution will vary over the set of pareto optimal solutions of the multi-objective case. Trade-off comparisons for different weighting values can be viewed graphically by completing a preliminary search to provide an estimation of the pareto boundary for the full set of production options. This method avoids using an exhaustive search, and allows the users to make informed decisions on what they consider to be optimal prior to picking an option.

4 Case Study

The case study focuses on a scenario in which a larger company is looking to out-source the production of a sub-assembly in either a temporary or long-term action. The product being used for the case study is currently being produced at a 3^{rd} party manufacturing company. As such we cannot include specifics such as design images due to their proprietary nature. This is consistent with the proposed framework in which the proprietary design information will be maintained during the initial matching process. Only after an agreement for production has been made will the design images become available to the candidate suppliers. This sub-assembly includes a turned housing with several pressed in bolt studs. While this sub-assembly is relatively simple in geometry, and only includes one non-purchased part, it does involve a variety of manufacturing operations. This processes diversity allows for validation of a more representative framework, while still maintaining a level of simplicity that was desired for the initial case study. In this scenario, it is important that the lead time is minimized, while maintaining a relatively high level of quality, and reasonable cost profile.

4.1 Manufacturing Process Decomposition

In order to create the production request for the case study, the manufacturing of the housing was broken down into six operations:

1. Casting
2. Exterior Turning and Boring
3. Drilling and Tapping
4. Plating
5. Interior Turning
6. Pressing of Studs

This breakdown assumes that similar operations that are done in series will be done at a single supplier, and most likely on a single machine. For this case study, this assumption ensures that similar operations such as drilling and tapping of multiple holes be confined to a single step. For alternative products, the assumption of multi-step manufacturing machines may not be feasible or desirable.

The required process definitions described in section 3.1 were created for each of the processes used in the case study. For example, the process definition for turning includes attributes for part length, max diameter, weight and material. For each operation the required attributes as defined by the process definitions were included to provide a sufficient basis for selecting candidate suppliers. Some of these parameters carry across multiple operations, while others are operation specific. Using this information a production request was created for 100 parts.

4.2 Database Population

The PaaS framework was populated with nine different fictitious suppliers. Of the nine suppliers, three were casting suppliers (two of which had available machining resources), four were machine shops, and two were plating companies. This population was meant to be representative of the suppliers that have the available capacity and capability to complete some or all of the requested production tasks; suppliers that would be sorted out prior to soliciting quotes were not entered for this simple case study.

From these nine companies, fifteen quotes were generated using the information in the production request, and submitted into the database. The casting and plating quotes were based on

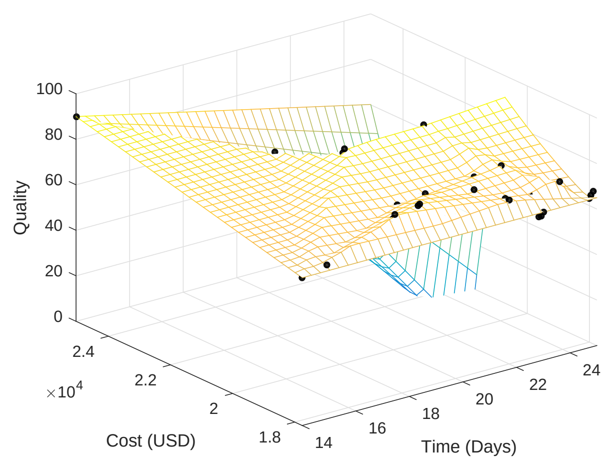

Figure 3. Production options for Case Study

Table 1. Case Study Variables

Variable	Value
Size of P	33 Production Options
ℓ	6 Production Steps
Range of $C(p)$	\$17760 - \$25039
Range of $T(p)$	14 - 25 Days
Range of $Q(p)$	52% - 90%
C_N	\$22054
T_N	14 Days
Q_N	78%

raw material cost in addition to the cost and time associated with tooling. Machining operations were estimated based on the time to complete the machining operation, which were found using the best practices in the Machinery's Handbook [14].

The populated quotes provide 33 production options that have been plotted in figure 3. This plot would not be generated for the customer because it requires an exhaustive search, but in this case the total number of options is small and can be viewed relatively easily in this format.

4.3 Optimization Results

The quotes were used to create a graph structure as described in section 3.4 and the preliminary search was carried out. Based on this preliminary search the output for the customer would include the ranges for each of the objectives which can be found by running the graph search to maximize and minimize each of the optimization parameters $C(p)$, $T(p)$, and $Q(p)$ individually. These values along with other variables from the case study are provided in Table 1. The customer would be provided with the plots in figure 4. These plots were generated by finding the normalizing constants C_N, T_N and Q_N using equations (8)-(10), varying the optimization weights at integer values over the range of -5 to 5 and running the graph search to find the optimal solution defined by equations (6) and (7) at each value. The resulting plots show the trade-offs based on different weightings. Using these plots the customer could make an informed decisions as to which option best suits their needs. At this point it would be up to the customer to review the specific details of the options they find suitable in this smaller subset of solutions. This process may include using filters such as range limits for a given variable.

In this case, the resulting number of options is quite small and the optimization relatively trivial given that the customer could view and consider each of the options in a reasonable

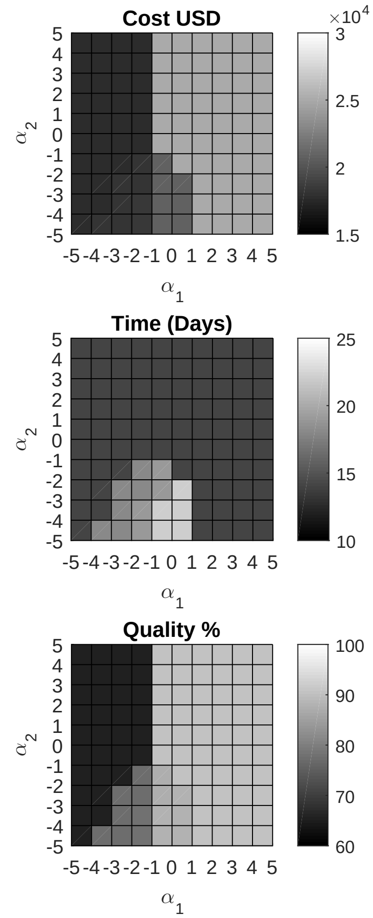

Figure 4. Optimal Solutions Under Varying Weights

amount of time. By increasing the number of quotes to 30, the total number of production options increases to 126. In this case, and in cases where the number of quotes is considerably larger, the optimization method presented in equations (6)-(10) provides a useful tool for narrowing the search to a smaller set of production options. Additionally, it allows for the optimization to be carried out without the use of an exhaustive search, focusing instead on finding enough information to describe the characteristics of the set and then implementing a graph search algorithm.

5 Conclusion

The proposed framework provides a platform that benefits both customers and suppliers. From a supplier standpoint, little effort is required to participate, while the returns can help to increase efficiency through the use of idle resources. The structure of the framework provides robust scalability both in number of participants and in supported processes. The back end structure allows for extensibility through modular software design and distributed computing. Process definitions allow for new processes to be added and updated over time. Security concerns for supplier information are handled by the design of the back-end structure. Customers looking to have something produced are automatically paired with potential suppliers and then given the tools to find the production option that best fits their needs. Customers are allowed to weight the optimization metrics of cost time and quality according to their preferences. Additional constraints to the optimization metrics are discussed and can be added at a later time. Because the resources would otherwise sit idle, small-scale manufacturing requests will likely benefit the most.

ACKNOWLEDGMENT

The authors would like to acknowledge National Science Foundation funding for this research through the Early Concept Grants for Exploratory Research (EAGER) program grant number NSF 1546036.

REFERENCES

[1] Manufacturers moving to be closer to consumers as new technologies and mass customisation shift global trends in manufacturing. [Online]. Available: http://www.colliers.com/en-gb/uk/insights/property-news/2014/0915-manufacturers-moving-closer-to-consumers-as-new-technologies-shift-global-trends, Accessed: 2016-04-10.

[2] Kull, H., 2015. *Mass Customization: Opportunities, Methods, and Challenges for Manufacturers*. friendsof ED, Apress, New York.

[3] Macia-Perez, F., Berna-Martinez, J. V., Marcos-Jorquera, D., Lorenzo-Fonseca, I., and Ferrandiz-Colmeiro, A., 2012. "A new paradigm: cloud agile manufacturing". *International Journal of Advanced Science and Technology,* ***45***, pp. 47–54.

[4] Mersch, H., and Epple, U., 2011. "Requirements on Distribution Management for Service-Oriented Automation Systems Chair of Process Control Engineering". *IEEE Emerging Technologies & Factory Automation*.

[5] Cachapa, D., Harrison, R., and Colombo, A., 2011. "Configuration of SoA-based devices in virtual production cells". *International Journal of Production Research,* ***49***(24), pp. 7397–7423.

[6] Moergestel, L., Puik, E., Telgen, D., and Meyer, J., 2015. "Implementing manufacturing as a service: A pull-driven agent-based manufacturing grid". *CEUR Workshop Proceedings,* ***1356***, pp. 172–187.

[7] Cândido, G., Colombo, A., Barata, J., and Jammes, F., 2011. "Service-oriented infrastructure to support the deployment of evolvable production systems". *IEEE Transactions on Industrial Informatics,* ***7***(4), pp. 759–767.

[8] Rauschecker, U., Meier, M., Muckenhirn, R., Yip, A., Jagadeesan, A., and Corney, J., 2011. "Cloud-based manufacturing-as-a-service environment for customized products". *eChallenges e-2011 Conference Proceedings*, pp. 1–8.

[9] Dotoli, M., Fanti, M., Meloni, C., and Zhou, M., 2006. "Design and optimization of integrated e-supply chain for agile and environmentally conscious manufacturing". *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans,* ***36***(1), pp. 62–75.

[10] Eng, S., Chew, E., and Lee, L., 2014. "Impacts of supplier knowledge sharing competences and production capacities on radical innovative product sourcing". *European Journal of Operational Research*.

[11] Hadoop. [Online]. Available: http://hadoop.apache.org, Accessed: 2016-03-17.

[12] Dijkstra, E. W., 1959. "A Note on Two Problems in Connexion with Graphs". *Numerische Mathematik ˜,* ***1***(1), pp. 269–271.

[13] Yen, J. Y., 1971. "Finding the K Shortest Loopless Paths in a Network Author". *Management Science,* ***17***(11), pp. 712–716.

[14] Oberg, E., 2012. "Machinery's handbook: a reference book for the mechanical engineer, designer, manufacturing engineer, draftsman, toolmaker, and machinist". In *Machinery's handbook*, C. J. McCauley, ed., 29 ed. Industrial Press, New York, pp. 1004–1327.