WHAT DOES MULTI-OBJECTIVE OPTIMIZATION HAVE TO DO WITH BOTTLENECK IMPROVEMENT OF PRODUCTION SYSTEMS?

Amos H.C. Ng[1], Jacob Bernedixen[1,2] and Leif Pehrsson[1,2]

[1]*Virtual Systems Research Centre, University of Skövde, Sweden*
[2]*Volvo Car Corporation, Sweden*

amos.ng@his.se

Abstract: Bottleneck is a common term used to describe the process/operation/person that constrains the performance of the whole system. Since Goldratt introduced his theory of constraint, not many will argue about the importance of identifying and then improving the bottleneck, in order to improve the performance of the entire system. Nevertheless, there exist various definitions of bottleneck, which make bottleneck identification and improvement not a straightforward task in practice. The theory introduced by Production Systems Engineering (PSE) that the bottleneck of a production line is where the infinitesimal improvement can lead to the largest improvement of the average throughput, has provided an inspirational and rigorous way to understand the nature of bottleneck. This is because it conceptually puts bottleneck identification and improvement into a single task. Nevertheless, it is said that a procedure to evaluate how the efficiency increase of each machine would affect the total performance of a line is hardly possible in most practical situations. But is this true?

In this paper, we argue how multi-objective optimization fits nicely into the theory introduced by PSE and hence how it can be developed into a practical bottleneck improvement methodology. Numerical results from a real-world application study on a highly complex machining line are provided to justify the practical applicability of this new methodology.

Keywords: Bottleneck Improvement, Production System Simulation, Multi-objective Optimization, Data Mining.

1. INTRODUCTION

Excellence in manufacturing is often a result of a combination of successive incremental investments in technology and improvements in equipment (Sim, 2001). But in complex production environments, where and what to invest and improve have been proven to be some very challenging tasks, even for seasoned production managers/engineers. In order to increase throughput and hence the operational expense of a company, Theory of Constraints (ToC), introduced by Goldratt in the 1980s (Goldratt, 1984), suggests that the key is to improve the constraint in its system. Like Lean Manufacturing and Six Sigma, ToC is another management theory that focuses on system improvement. But unlike Lean and Six Sigma, ToC specifically emphases that the performance of the entire value chain is limited by the strength of its weakest link (Nave, 2002). ToC states that in any system, there is a single constraint that is limiting the system’s overall ability to achieve a better goal. Therefore, the ToC procedure starts with identifying and exploiting the constraint, and then “elevating” it, in order to improve the entire system. As a matter of continuous improvement, this cycle is continuously repeated when the constraint has shifted to another part of the system. In a production system, such a constraint is very often referred to as the bottleneck.

In the literature, there are various methods for detecting both momentary and steady-state bottlenecks. These include utilization of machines (Hopp and Spearman, 2000), blocking and starving patterns (Kuo, et al., 1996), data-driven approach (Li, 2009), shifting bottleneck detection (Roser, et al., 2001), multiple bottlenecks (Aneja and Punnen, 1999) as well as a method based on inter-departure time and failure cycle data (Sengupta, et al., 2008). Nevertheless, all of these existing methods have the same deficiency: even if the overall constraint of the system can be identified down to a specific workstation or machine in the system, there is not enough information provided for determining what improvement action(s) has to be taken. This could, in some cases, be a serious disadvantage because local improvement of the wrong parameter may actually degrade the performance of the whole system (Ignizio, 2009). Another issue is how to interpret the results of the above-mentioned methods when they are applied to complex production systems with buffers, parallel and serial flows, feedback loops, operational logic, rework, and variant-specific operations. It is likely that the more complex the system, a more effective method is needed in finding the right combination of improvement actions to enhance the performance of the system, especially within a limited budget.

This paper introduces a completely new way to improve bottlenecks of production systems, through a seemingly irrelevant method, namely, multi-objective optimization (MOO). While in general, optimization can be used to find a solution to improve any bottleneck by formulating the improvement problem as an optimization problem. But in practice this is not straightforward as any decisions in an improvement project would become a matter of resolving conflicting objectives and require some compromise/trade-off. As a matter of fact, there is almost always some economical budget and/or technical constraint that limit the possibility to improve a bottleneck. Unlike single-objective optimization that the goal is to seek a single optimal solution to maximize or minimize a single optimization objective function, MOO aims at seeking a set of optimal trade-off solutions with respect to multiple (≥2) conflicting objectives (Deb, 2004). The method proposed in this paper is based on finding the most beneficial improvement combinations for a production line through the use of simulation-based multi-objective optimization. The input variables of the optimization problem are the local system parameters, such as machine cycle time, machine availability and mean down time that are considered to be possible to be improved. By simultaneously minimizing the total number of such changes and maximizing throughput of the system, solutions with the most beneficial throughput improvement, given the fewest number of system changes, will be found in a single optimization run.

This paper provides the theoretical basis to show why such a MOO-based method fits well as a bottleneck improvement methodology, which we argue can produce more promising results than ToC and Lean Manufacturing. It also provides the numerical results from a real-world application study on a highly complex real-world machining line in an automotive manufacturer, to justify the practical applicability of the methodology. The rest of the paper is organized as follows. We have a closer look at some important definitions of bottleneck in Section 2. Section 3 introduces MOO and followed by Section 4 on how MOO fits nicely as an automated bottleneck identification and improvement method. The industrial application study is described and analyzed in Section 5. The paper is concluded with some comparisons to Lean manufacturing and ToC.

2. BOTTLENECKS: A MATTER OF DEFINITIONS

Bottleneck is a common term used not limited to production systems but any types of systems or processes in daily lives. By common sense, a bottleneck is a process/operation/person which constrains the performance of the whole system. It is by this common sense understanding of bottleneck that ToC is developed. Nevertheless, as a matter of fact, there are various definitions of bottleneck, which making the task of identifying the bottleneck process/operation/person not that straightforward in practice.

2.1. Factory Physics

In their award-winning textbook, “Factory Physics”, Hopp and Spearman (2000) define the capacity of a system as “the maximum average rate at which entities can flow through the system.” In this regard, utilization of a machine/station, um, is defined as:

$$u_m = \frac{r_m}{c_m} \qquad m = 1,2,\dots,N$$

Where r_m is the arrival rate into the machine and c_m is the machine capacity, and N is the total number of machines.

With this understanding of utilization, Factory Physics states that the bottleneck of a system is the station with the highest utilization. Recall that in queuing theory, the queuing time (QT) of a system will approach ∞ when u_m

→1 because *QT α u/u-1*. Therefore, a basic principle of production system is that the throughput of a system is always less than its capacity. In general, the common understanding of practitioners in measuring average u_m in order to detect bottleneck conforms to this definition. In practical situations, both in real shopfloor and in simulation models, it is possible to measure the capacity of each station and the average rate parts flow into it, in order to determine u_m. But this kind of data measurement is actually seldom implemented, both in real and simulated production lines. Nonetheless, in the literature, there are other more advanced approaches to define and determine bottleneck.

2.2. Toyota's Shifting Bottleneck

Shifting bottleneck detection was developed by the Toyota Software Laboratory in early 2000 (Roser, et al., 2001), based on a similar concept introduced by Adam et al. (1988). It defines the machine with the longest average active period as the bottleneck. During production, the machines in a production line alter state over time. The different states: working, failed, setup, waiting, and blocked are logged together with information about breaks and unplanned activities in order to perform the shifting bottleneck analysis. The different states are grouped into active and in-active periods, in which the states working, failed, and setup are considered active. The momentary bottleneck is the machine with the longest active period. If there is an overlap between two consecutive bottlenecks, the overlap period is marked as a shifting bottleneck period and both bottlenecks are considered momentary bottlenecks during the shifting period, see Fig. 1.

This method was further developed into a shifting bottleneck detector (Roser, et al., 2002) and was proven to work in almost all kind of resources such as machines, workers and AGVs. The probability of a resource to be a bottleneck is done by determining the percentage of time that has been sole and shifting bottleneck throughout the measuring period. For analysis, the method is also able to identify secondary bottlenecks. However, it is said that, "the shifting bottleneck detection method has the one flaw of being slightly more difficult to implement" (Roser, et al., 2003) than other methods.

Fig.1. Active periods in shifting bottleneck detection.

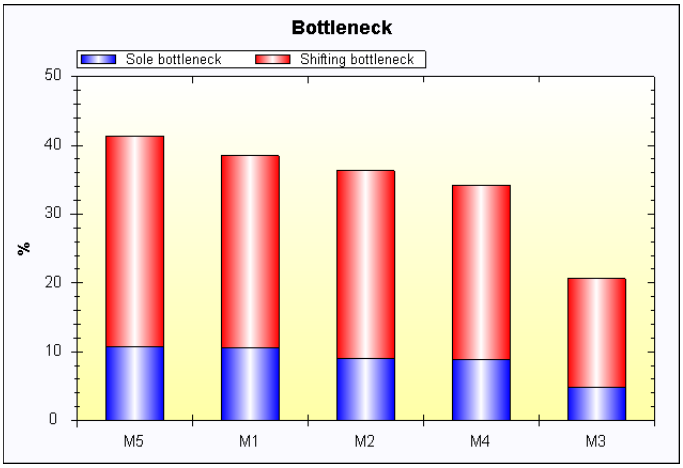

Fig.2. Sole and shifting bottleneck (plotted in FACTS Analyzer).

2.3. Production Systems Engineering

The PSE textbook (Li and Meerkov, 2009) provides some rigorous concepts to understand the nature of bottlenecks. In terms of continuous improvements, they define two approaches: (1) constrained improvability, addressing the situation that a production system can be improved by re-allocating its existing resources, like re-allocating workloads and buffer capacities; (2) unconstrained improvability is about identifying and then eliminating the bottleneck, by allocating new, additional resources, including additional buffer capacity, machine improvement through reducing processing times or even new machines, etc. In terms of unconstrained improvability, PSE further defines the concept of *BN-m* and *BN-b*, meaning bottleneck machines and bottleneck buffers, respectively. Assuming *TH* is the overall steady-state throughput of the production system, as measured as the average number of parts per unit of time coming out from the last machine. A machine m_i of a Bernoulli production line, with the probability of failure $1-p_i$ (i.e. probability of up is p_i), is the *BN-m*, if the infinitesimal improvement of p_i can lead to the largest improvement of *TH*. Mathematically,

$$\frac{TH(p_1,\dots,p_i+\partial p_i,\dots p_{N,}b_1,\dots,b_{N-1})}{\partial p_i} > \frac{TH(p_1,\dots,p_j+\partial p_j,\dots p_{N,}b_1,\dots,b_{N-1})}{\partial p_j} \qquad \forall j \neq i$$

Note that here *TH* is a function of the design/decision variables of the line represented by a vector *x*. For a Bernoulli line, as the one illustrated in Fig. 3 with total number of machines *N=5*, its configuration can be represented as $x = [p_1, \dots, p_N, b_1, \dots, b_{N-1}]$. For this example line, *x* = [0.92, 0.9, 0.95, 0.95, 0.92, 1, 2, 2, 1].

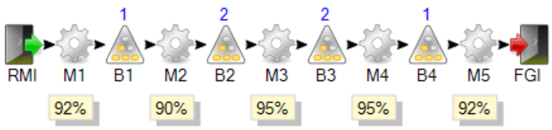

Fig. 3. A simple flow line with N=5.

Likewise, the concept of bottleneck buffer, *BN-b*, of a Bernoulli line can be defined as the buffer that leads to the largest increase of *TH*, if its capacity is increased by 1 (in most cases, buffer capacity is a discrete decision variable for discrete manufacturing), as compared with the increasing in other buffers:

$$TH(p_1,\dots,p_{N,}b_1,\dots,b_i+1,\dots,b_{N-1}) > TH(p_1,\dots,p_{N,}b_1,\dots,b_j+1,\dots,b_{N-1}) \qquad \forall j \neq i$$

In this way, PSE has provided an inspirational and rigorous concept for understanding the nature of bottleneck because unlike the other definitions, it conceptually puts bottleneck identification and improvement into a single task. Nevertheless, even the PSE authors, Li and Meerkov (2009) have stated that a procedure to evaluate how the efficiency increase of each machine would affect the total performance of a line "is hardly possible in most practical situations". But does there exist some way to make this possible in practical situations? Before answering this question, we shall first introduce the concept of multi-objective optimization.

3. MULTI-OBJECTIVE OPTIMIZATION

As the name suggests, MOO involves optimizing more than one objective simultaneously. Consider an MOO problem in its general mathematical form according to Deb (2004):

$$\text{Minimize/Maximize } f_m(\boldsymbol{x}), \qquad m = 1,2,\dots,M$$
$$\text{Subject to } g_j(x) \geq 0, h_k(x) = 0,$$
$$\mathrm{j} = 1,2,\dots,\mathrm{J};\, k = 1,2,\dots,\mathrm{K}$$
$$\text{With respect to } x = (x_1, x_2, \dots, x_n)^T, \text{ where } x_i^L \leq x_i \leq x_i^U \text{ and } i = 1,2,\dots n.$$

Here, $f_m(\boldsymbol{x})$ represents the objectives, which can be minimized and maximized with *x* as a solution vector (or simply a solution), consisted of *n* decision variables within their respective lower bounds (x_i^L) and upper bounds (x_i^U). The solutions have also need to satisfy the inequality constraints, $g_j(\boldsymbol{x})$, and equality constraints, represented by $h_k(\boldsymbol{x})$. In many MOO applications, where the objectives $f_m(\boldsymbol{x})$ are in conflict with each other, finding a single best optimal solution is impossible because improving one objective would deteriorate the others. This gives rise to the concept of Pareto-optimality, describing the set of optimal solutions which are the best trade-offs with respect to $f_m(\boldsymbol{x})$. In order to determine such a set of optimal solutions, popularly known as Pareto-optimal solutions, the concept of dominance is commonly used by many MOO algorithms:

Definition 1. A solution $\boldsymbol{x_1}$ is said to dominate the other solution $\boldsymbol{x_2}$, if both of the following two conditions hold true:

1. The solution $\boldsymbol{x_1}$ is no worse than $\boldsymbol{x_2}$ in all M objectives. So without loss of generality, if we consider a problem of minimizing all $f_m(\boldsymbol{x})$ objectives, then $f_m(\boldsymbol{x_1}) \leq f_m(\boldsymbol{x_2})$, $\forall$ $m = 1,2, \ldots, M$..
2. The solution $\boldsymbol{x_1}$ is strictly better than $\boldsymbol{x_2}$ in at least one objective, i.e. $\exists\, j \in (1,2,\ldots,M)$ such that $f_j(\boldsymbol{x_1}) < f_j(\boldsymbol{x_2})$.

How this seemingly irrelevant concept of dominance can be effectively used to address bottleneck improvement will become apparent when we show how a system improvement problem can be formulated as an MOO problem. So, the key question raised in this paper, "*what does MOO have to do with bottlenecks improvement?",* will be explained in the next section.

4. MOO FOR BOTTLENECK IMPROVEMENT

The basic idea proposed in (Pehrsson, et al., 2001a, b; 2013) was based on an observation that many decision-making situations in production system improvement projects can be effectively formulated into an MOO problem. While the primary objective is usually related to a key performance measure, such as system throughput or total cycle time, the novelty of the approach proposed is on formulating the investments needed to improve various attributes of the system as a summation function to represent the second objective of the MOO problem. In a more generalized case, if the system throughput (*TH*) is the primary objective for the improvement, so that $f_1(x)= TH(x)$, then we can define the total number of changes, i.e. improvement actions, as the second objective function, $f_2(x)$, in the MOO problem. Precisely, if we consider three types of discrete, two-level decision variables that can either be set to the system's original value or to an improved value, caused by an improvement action, e.g. availability increase from α_i to $\alpha_i+\partial\alpha_i$. Then, we can formulate the second objective, $f_2(x)$, in the MOO problem mathematically as a summation function of improvements:

$$f_2(x) = min\left\{\sum\nolimits_{i=1}^{N} \hat{\alpha}_i + \sum\nolimits_{i=1}^{N} \hat{\beta}_i + \sum\nolimits_{i=1}^{N} \hat{\gamma}_i\right\}$$

Where:

$$\hat{\alpha}_i = \begin{cases} 0 & if\ \alpha_i\ remains\ to\ be\ \alpha_i \\ 1 & if\ \alpha_i\ is\ improved\ by\ \partial\alpha_i \end{cases}$$

Similarly, $\hat{\beta}_i = 0$, if the processing time of machine i is unchanged. Otherwise, $\hat{\beta}_i = 1$, if it's cycle time is reduced (improved) from β_i to $\beta_i+\partial\beta_i$. In the same manner, $\hat{\gamma}_i = 1$, if MDT of machine i is improved by $\partial\gamma_i$.

With such a formulation, the second objective function is a discrete function varies in the range [0, $3N$] because the maximum number of changes can only be $3N$. Fig. 4 illustrates this bi-objective problem in a graphical way, showing how the Pareto-optimal solutions in the objective space can be used for supporting the decision making in choosing the optimal (minimal) changes in order to achieve the maximum throughput. By using this figure and understanding the concept of dominance as described above, several important concepts are revealed:

- The original throughput, TH_0, represents the current condition when no improvement has been made.
- If the target condition (some readers may notice that this term is borrowed from Toyota Kata by Rother (2009)) is TH_t, then the 'optimal' number of changes required for improving the line from TH_0 to TH_t is 4, denoted by the solution x_1 which consists of the required improvement actions at the right positions to achieve this. All other solutions with $f_2(x)$, e.g. x_2 and x_3 are 'inferior' to x_1 because with the same number of changes, the throughputs achieved are lower.
- TH_{max} represents the maximum practical *TH* that the line can achieve no matter how many changes are implemented. Note that this is called maximum practical *TH* because in theory, if processing times of the workstations can be unlimitedly reduced, then the system throughput can be also unlimitedly increased (recall Little's Law (Little, 1992)). Since reducing physical processing times can only be done to a limited extent as well as β_i and γ_i are always practically <100% and >0 respectively, there is always a maximum practical TH_{max} in a real production system.

- If the target is to reach the maximum attainable TH_{max} of the line, then C_{max} improvement actions at the right places have to be implemented. In other words, (C_{max}, TH_{max}) denotes the optimal (combined) improvements that make the line achieves its maximum practical TH_{max}.
- Any solutions, x_i, beyond (C_{max}, TH_{max}) with $f_1(x_i) = TH_{max}$ and $f_2(x_i) > C_{max}$ are dominated by (C_{max}, TH_{max}) because they require more, redundant improvement actions to reach TH_{max}.

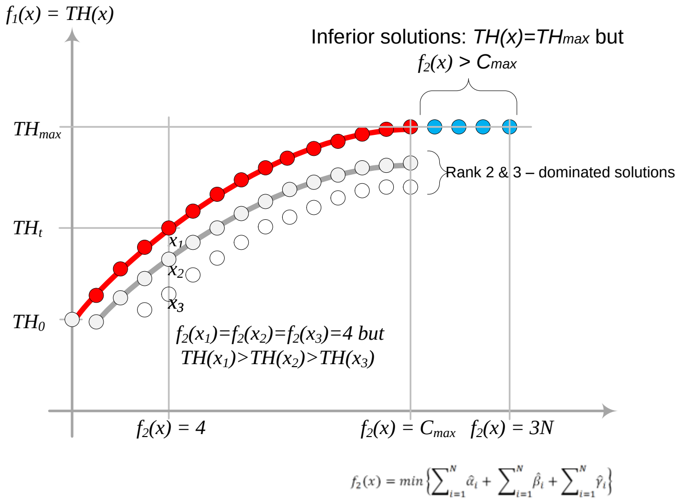

Fig. 4. MOO for system improvement.

With such an MOO approach, all attainable *TH* of the system, through various combinations of improvement actions, can be sought in a single optimization run. If a decision maker has decided for a certain target TH_t, the optimality of the corresponding solution found on the Pareto front is ensured by the dominance relation. Take the example as illustrated in Fig. 4, x_1 dominates x_2 and x_2 dominates x_3 because of the relation $f_2(x_1)=f_2(x_2)=f_2(x_3)=4$ and $TH(x_1)>TH(x_2)>TH(x_3)$. Put this back into *Definition 1*, x_1 is no worse than x_2 and x_3 in $f_2(x)$ but is strictly better than them in $f_1(x)$. And if we think of this relation in the light of the PSE definition of bottleneck, x_1 then represents the optimal improvement combinations of x that can lead to the largest improvement of *TH*. On one hand, this approach deviates from the original definition of PSE because the optimization does not consider only each infinitesimally single change at a position (e.g. m_i) but is more powerful in the sense that it can determine the largest improvement of *TH* with the optimal improvement combinations. More concretely, unlike in PSE that the determinations of BN-m and BN-b are separated (Li and Meerkov, 2009), the MOO approach can provide a more computationally-effective and accurate approach because combinations of improvements and buffer allocations can be evaluated simultaneously in the same optimization run. This advantage shall not be underestimated because the complex interactions between the buffers and the machines can cause joint effect in the system performance, which cannot be detected if the changes in the buffer allocation and other resource allocation are evaluated separately. On the other hand, the PSE definition provides a rigorous way to identify exactly where is the bottleneck, which may not be as straightforward when using the proposed MOO approach. But if multiple changes are essential to achieve the target condition desired by the production manager, then it is argued that MOO provides a more practical solution than PSE because possibly multiple bottlenecks have to be overcome, which making the identification of a single bottleneck less meaningful. Having said that, the indication of the exact bottleneck position(s) can be equally important than just improving the system. In order to acquire some deeper knowledge about the production system under study, an innovative post-optimality analysis proposed in (Pehrsson, 2013) is in order. Because of the limited space, we refer readers to his thesis for the details of the post-optimality analysis methods as well as for more experimental results on proving the validity of the MOO approach. In this paper, the practical applicability of the methodology to determine the bottleneck position is demonstrated through the results from an application study in a highly complex real-world machining line in a Swedish automotive manufacturer.

5. A COMPLEX INDUSTRIAL APPLICATION

The complex production line under study is the same one first presented in (Siegmund, et al., 2012), but in the current study the simulation model is re-built with FACTS Analyzer (Ng, et al., 2008), or simply FACTS hereafter, developed at the University of Skövde. There are two main features that distinguish FACTS from other production system simulation software: (1) rapid modelling, no programming, easy to learn and use; (2) tightly integrated with powerful multi-objective optimization algorithms, making optimization of production systems straightforward. This latter feature renders any FACTS models to flexibly cope with a list of generic optimization parameters, instead of like the case in many commercial discrete-event simulation software in which tailored optimization parameters have to be added, usually in form of programming code, to the models. The snapshot of the FACTS model in Fig. 5 illustrates the complexity of the production system. The production line includes several parallel sections, assembly stations, machines with several operations, portal cranes for complex variants handling, etc. The size and complexity of the line makes it extremely hard to locate what to improve, let alone the effect of each improvement. With the objectives to simultaneously maximize the system throughput as well as to minimize the number of improvements, to apply simulation-based MOO is in order. The production engineers in charge of the production line provided improvement options like reduced processing times (per variant where applicable), increased availabilities, and reduced mean times to repair. The levels of the improvement (from the original value) are varied between different stations. Table 1 lists the number of improvement variables of each type and also the range of the improvements of that type, e.g. the processing time reduction ranges from only 0.2 % for one station to 41.8 % for another station. That sums up to 464 improvement alternatives represented in the optimization problem as binary Multiple Choice Set (MCS) variables (Bernedixen and Ng, 2014). That is equivalent to $2^{464} \approx 4.8E+139$ possible improvement combinations.

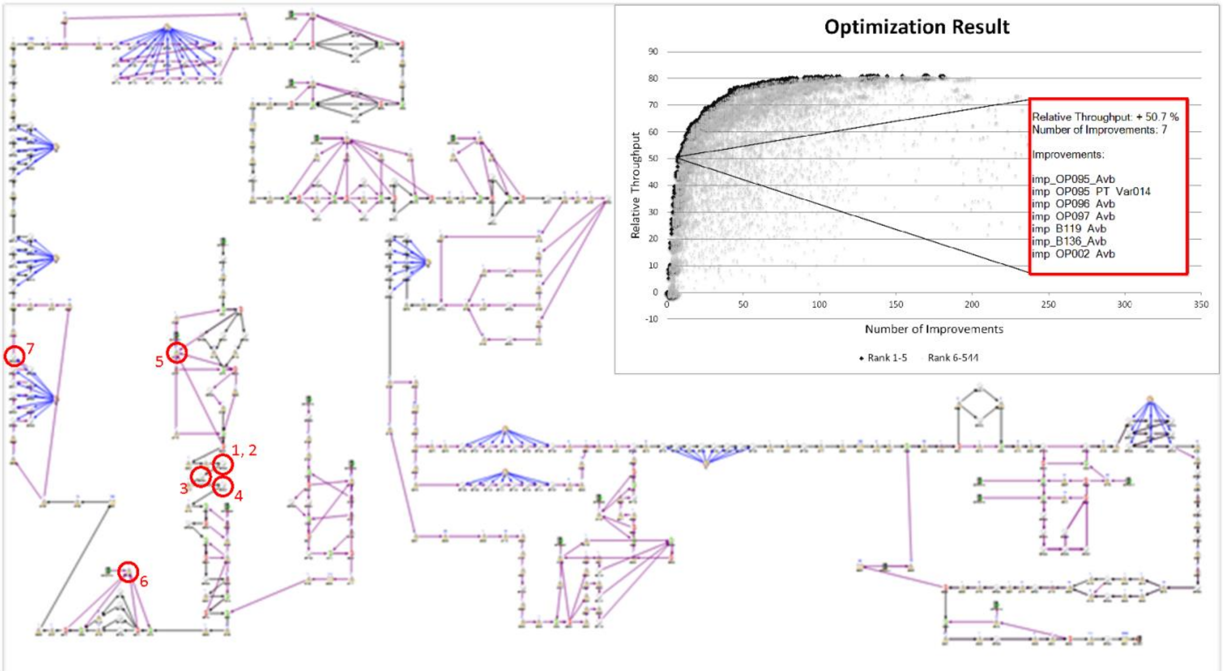

Fig. 5. FACTS model of the complex machining line and its MOO results.

Table 1. Improvement details.

Type of improvement variable	**Number of variables of this type**	**Range (min-max)**
Processing time	317	-0.2% to -41.8%
Availability	82	+0.1% to +23.8%
Mean time to repair	65	-5.5% to -92.1%

The optimization results generated from the well-known NSGA-II algorithm developed by Deb et al. (2002) for this MOO problem are also shown in the data plot in Fig. 5. In order to conceal the real data of the company, here *TH* is expressed as relative change in percent from the initial state with no improvements (i.e. TH_0). It can be seen that these improvements can improve the *TH* significantly (over 80 %) and a substantial improvement (about 50 %) can be achieved with only 7 improvements.

Inspired by a specialized innovization procedure to give a rank order for sports player selection (Ahmed, et al., 2013), the importance of the different improvements among the best solutions (black dots in the sub-plot in Fig. 5) is determined by a simple frequency analysis. In this case, the best solutions have been selected as the rank 1 to 5 solutions, ranked using the fast non-dominated sorting described in (Deb, et al., 2002). The frequency analysis simply calculates the frequency of each improvement among these best solutions. The 20 most frequent ones are shown in Fig. 6. Here, operation *OP095* stands out as the most influencing station – 5 out of the top 20 most frequent improvements are related to *OP095*, one for the availability (Avb) and the other 4 for the processing times (PT) for processing different product variants on *OP095*.

In-line with the concept of automated innovization that various data mining techniques can be used to perform post-optimality analyses on the generated MOO datasets (Ng, et al., 2011), more advanced data mining techniques have been applied on the rank 1-5 solutions in order to extract patterns/rules/knowledge from the optimal solutions. Particularly, Fig. 7 shows the bar chart indicating the importance of both individual improvement actions and their interactions (highlighted in red), using the Apriori Algorithm of Sequential Pattern Mining (SPM) (Agrawal and Srikant, 1995).

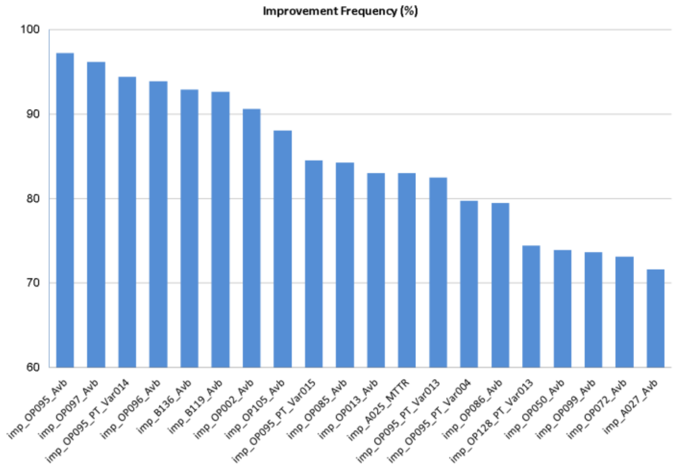

Fig. 6. The 20 most frequent (%) improvement actions among the best MOO solutions.

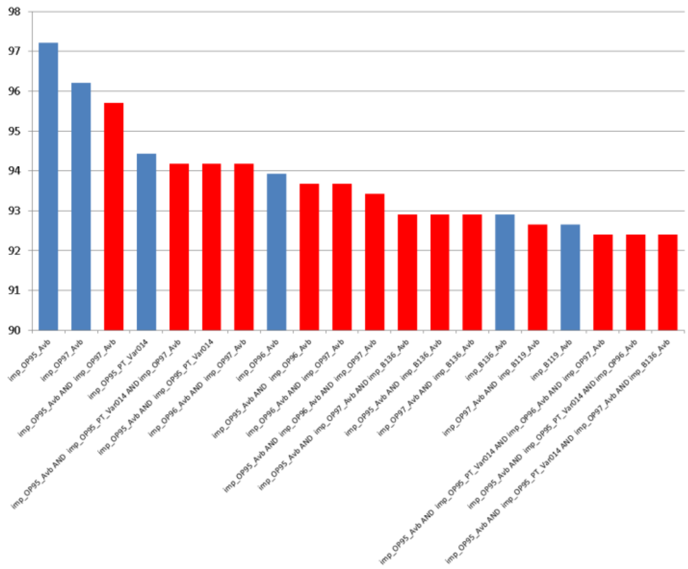

Fig. 7: Significance (%) of improvement actions and their interactions generated by running sequential pattern mining on the MOO data.

While this type of plot resembles Pareto chart generated from DoE (Faget, et al., 2005), it should be noticed DoE does not consider the effect of minimizing the number of changes as the proposed MOO-based approach. The SPM results provide a cross-validation test for the simple frequency analysis to pinpoint exactly that *OP95, OP96, OP97, OP136 & OP119* are the "problematic" stations. In addition, the high importance about their improvement combinations, e.g. availability of *OP95 AND OP96*, as appears as the third highest significance in Fig. 7, gives a strong indication that bottleneck cannot be addressed effectively using the "detect, exploit and elevate" cycles, each on a single bottleneck, as proposed in ToC.

6. DISCUSSIONS AND CONCLUSIONS

This paper has illustrated how MOO can be used to generate the optimal improvement combinations for bottleneck of production systems. By the definition of PSE, a bottleneck is the machine that is most sensitive to throughput from the system compared to the improvement of a machine which is not the bottleneck. The magnitude of a bottleneck can be defined as the magnitude of the machine's throughput effect related to the system's throughput. The MOO-based method proposed in this paper has offered the opportunity for a decision maker in continuous system improvement to gain insight into the system's behavior, the nature of the constraints, and the optimal improvement actions required to achieve the desired system performance. In contrast to ToC, which relies on repeated cycles of "detect, exploit and elevate" and Toyota Kata, which relies on defining step by step the experimentation of single improvement that can lead towards the target condition, the proposed MOO method is powerful, in the sense that all attainable *TH* levels and the corresponding optimal combinations of improvement actions are generated automatically in a single optimization run.

One may notice that we mentioned about investment cost and limited budgets in the beginning of this paper, but they are not further discussed. As a matter of fact, in a practical application, if the decision maker can relate every change to a cost figure, then the MOO problem can be formulated readily as a cost of changes versus *TH* problem. Actually, this has already been done in our previous studies (Pehrsson, et al., 2001a, b; 2013). While this may be a convenient option for the decision maker, it is argued in the current paper that for the sake of acquiring deeper knowledge on the system under study, an MOO run for generating a "number of changes versus TH" data plot is still necessary and informative for identifying which are the most influencing decision variables. Those changes might be prohibitively expensive or even technically impossible to improve, unless, for instance, replaced with a new piece of equipment. Nevertheless, there is no doubt that for any production line managers/engineers, they are keen to know accurately where and what is their real bottleneck.

7. ACKNOWLEDGEMENT

This work was initiated during the FFI-HSO project (2009-2012) funded by VINNOVA. The research is continued within the Apply-IT research school, operated at the University of Skövde, via the funding of KKS and Volvo Car Corporation. The authors gratefully acknowledge their financial supports over the years.

REFERENCES

Adams, J., E. Balas and D. Zawack. (1988). The Shifting Bottleneck Procedure for Job Shop Scheduling, *Management Science*, **Vol. 34**, No. 3, 391-401.

Agrawal, R. and R. Srikant (1995). Mining sequential patterns, *Proceedings of the Eleventh International Conference on Data Engineering*, 6-10 March, 3-14.

Ahmed, F., K. Deb and A. Jindal (2013). Multi-objective optimization and decision making approaches to cricket team. *Applied Soft Computing*, **13**, 402-414.

Aneja, Y.P. and A.P. Punnen (1999). Multiple Bottleneck assignment problems. *European Journal of Operational Research*, **112**, 167-173.

Bernedixen, J. and A.H.C. Ng (2014). Practical production systems optimization using Multiple-Choice Sets and Manhattan distance based constraints handling. *The 12th Industrial Simulation Conference*, 11-13 June, EUROSIS.

Deb, K., A. Pratap, S. Agarwal and T. Meyarivan (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transaction on Evolutionary Computation*, **Vol. 6**, No. 2, 181-197.

Deb, K. (2004). *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons Ltd.

Faget, P., U. Erikson and F. Herrman (2005). Applying discrete event simulation and an automated bottleneck analysis as an aid to detect running production constraints. *Proceedings of the 2005 Winter Simulation Conference*, 4-7 December, Orlando, FL, USA. 1401-1407.

Goldratt, E.M. (1984). *The Goal: A Process of Ongoing Improvement.* North River Press.

Hopp, W.J. and M.L. Spearman (2000). *Factory physics: foundations of manufacturing management*, 2nd ed., Irwin McGraw-Hill Higher Education, Burr Ridge, IL.

Ignizio, J.P. (2009). *Optimizing Factory Performance: Cost-effective ways to achieve significant and sustainable improvements*. McGraw-Hill, USA.

Kuo, C.-T., J.-T. Lim and S.M. Meerkov (1996). Bottlenecks in serial production lines: A systematic approach. *Mathematical Problems in Engineering*, **2**, 233-276.

Li, L. (2009). Bottleneck detection of complex manufacturing systems using a data-driven method, *International Journal of Production Research*. **Vol. 47**, No. 24, 6929-6940.

Li, J. and S.M. Meerkov (2009). *Production Systems Engineering*, Springer.

Little, J.D.C. (1992). Are there 'Laws' of Manufacturing, *Manufacturing Systems: Foundations of World-Class Practice*, J. A. Heim and W. D. Compton (eds.), National Academy Press, Washington, D.C., 180-188.

Nave, D. (2002). How to compare six sigma, lean and the theory of constraints, *Quality Progress*, 35, 73-80.

Ng, A.H.C., J. Svensson and M. Urenda (2008). Multi-Objective Simulation Optimization for Production Systems Design using FACTS Analyzer, *Proceedings of the 2'nd Swedish Production Symposium (SPS'08)*, Stockholm, Sweden, 2-3 November.

Ng, A.H.C., C. Dudas and K. Deb (2011). Simulation-Based Innovization using Data Mining for Production Systems Analysis. In *Evolutionary Multi-objective Optimization in Product Design and Manufacturing*. L. Wang, A. Ng, K. Deb (eds), Springer, 401-430.

Pehrsson, L. (2013). *Manufacturing Management and Decision Support using Simulation-based Multi-objective Optimisation*. PhD Thesis, De Montfort University, Leicester, UK.

Pehrsson, L., A.H.C. Ng and J. Bernedixen (2011a). Multi-objective production system optimisation with investment and running costs. In *Evolutionary Multi-objective Optimization in Product Design and Manufacturing*, L. Wang, A. Ng, K. Deb (eds.), Springer, 431-453.

Pehrsson, L., A.H.C. Ng and J. Bernedixen (2011b). Multi-objective production system optimisation including investment and running costs. *Proceedings of the 4th Swedish Production Symposium*, 3-5 May 2011, Lund, 55-62.

Pehrsson, L., A.H.C. Ng and D.J. Stockton (2013). Industrial Cost Modelling for Multi-Objective Optimisation of Production Systems. *International Journal of Computer and Industrial Engineering*, **Vol. 66**, Issue 4, 1036-1048.

Roser, C., M. Nakano and M. Tanaka (2001). A practical bottleneck detection method, *Proceedings of the 2001 Winter Simulation Conference*, 949-953.

Roser, C., M. Nakano and M. Tanaka (2002). Shifting bottleneck detection, *Proceedings of the 2002 Winter Simulation Conference*, **Vol. 2**, 1079-1086.

Roser, C., M. Nakano and M. Tanaka (2003). Comparison of bottleneck detection systems for AGV systems, *Proceedings of the 35th Winter Simulation Conference*, New Orleans, Louisiana, USA, 1192-1198.

Rother, M., (2009). *Toyota Kata: Managing People for Improvement, Adaptiveness and Superior Results*. McGraw-Hill.

Sengupta, S., K. Das and R.P. VanTil (2008). A new method for bottleneck detection, *Proceedings of the 2008 winter Simulation Conference*, 7-10 December, Hotel Intercontinental Miami, FL, USA. 1741-1745.

Siegmund, F., J. Bernedixen, L. Pehrsson, A.H.C. Ng and K. Deb (2012). Reference Point-Based Evolutionary Multi-Objective Optimization for Industrial Systems Simulation. *Proceedings of the 2012 Winter Simulation Conference*, Berlin, Germany, Dec.9-12, IEEE, 1-11.

Sim, K.L. (2001). An empirical examination of successive incremental improvement techniques and investment in manufacturing technology, *International Journal of Operations & Production Management*, **21**, 3, 373-399.