



HAL
open science

A Novel Metric for Determining the Constraining Effect of Resources in Manufacturing Simulation

Brian Patrick Kernan, Andrew Lynch, Con Sheahan, Dang Tung

► **To cite this version:**

Brian Patrick Kernan, Andrew Lynch, Con Sheahan, Dang Tung. A Novel Metric for Determining the Constraining Effect of Resources in Manufacturing Simulation. *International Journal of Production Research*, 2010, pp.1. 10.1080/00207543.2010.481643 . hal-00605258

HAL Id: hal-00605258

<https://hal.science/hal-00605258>

Submitted on 1 Jul 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



A Novel Metric for Determining the Constraining Effect of Resources in Manufacturing Simulation

Journal:	<i>International Journal of Production Research</i>
Manuscript ID:	TPRS-2009-IJPR-0748.R1
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	18-Jan-2010
Complete List of Authors:	Kernan, Brian; University Limerick, Enterprise Research Centre Lynch, Andrew; University of Limerick, Enterprise Research Centre Sheahan, Con; University of Limerick, Manufacturing and Operation Department Tung, Dang; University of Limerick, Manufacturing and Operation Department
Keywords:	SIMULATION, BENCHMARKING
Keywords (user):	



A Novel Metric for Determining the Constraining Effect of Resources in Manufacturing via Simulation

Brian Kernan, Andrew Lynch, Dang Tung, Con Sheahan

Manufacturing Operations & Engineering, University of Limerick, Limerick, Ireland

Brian Kernan, Engineering Research Building, University of Limerick

Andrew Lynch, Engineering Research Building, University of Limerick

Dr. Dang Tung, Engineering Research Building, University of Limerick

Dr. Con Sheahan, Engineering Research Building, University of Limerick

This paper provides a novel method for determining the constraining effect of resources in a manufacturing system using discrete event simulation. Traditionally manufacturing systems are constrained by one or more bottlenecks. Eliminating or mitigating the bottleneck will speed up the system throughput. However, bottlenecking resources generally only refer to machines, and primarily focus on flow-shops not job-shops. One important resource we believe that is often overlooked is workers and their associated skills, and we propose that a particular skill could be flagged as a bottleneck resource. We define new metrics known as Resource Constraint Metrics (RCM) for measuring the constraining effect of a resource on the entire manufacturing system. These metrics are flexible and differentiate between the constraining effects of machines and their requested skills. The metrics can also deal with complex workflows with alternative routing, alternative resources, calendars (a necessary consideration when dealing with workers), worker performance, and multiple modes of operation of machines (e.g. run, setup, and maintenance). The use of RCMs in simulation aids in real-world decision-making, by determining which resource should be focussed on and improved to reduce the overall system constrainedness. This will have the effect of increasing throughput or at least providing the capacity for increased throughput.

Keywords: manufacturing, enterprise, resource, machine, skill, starvation

1. Introduction

The work presented in this paper is founded primarily on the area of research of bottleneck detection in manufacturing simulation. There are numerous definitions as to what constitutes a bottleneck (Lawrence 1994). In Roser's paper (Roser 2002) a bottleneck is defined as a stage in a production system that has the largest effect on slowing down or stopping the entire system. Finding the bottleneck is no trivial task, and Cox and Spencer (Cox 1997) for example simply recommend that '...the best approach is often to go to the production floor and ask knowledgeable employees.' There are however a number of systematic methods available to locate the bottleneck for production systems however. One approach is to measure the utilisation of the different machines of the production system (Law 2000). The machine with the highest utilisation is considered to be the bottleneck. However, the utilisations of different machines are often very similar, so relatively long simulation run-times may be required to determine the system bottleneck with a high degree of confidence. Another frequently used method analyses the queue lengths of the machines in the production systems. In this method, either the queue length or the waiting time is determined, and the machine with the longest queue length or waiting time is considered to be the bottleneck. This method of detection is similar to looking at machine utilisation, since a machine with a large queue should also have commensurately high utilisation.

The shifting bottleneck method (Roser 2002) further distinguishes between a momentary bottleneck, describing the bottleneck at any given point in time, and an average bottleneck, describing the bottleneck behaviour over a selected period of time. The shifting bottleneck approach is based on the theory of constraints (Blackstone 2001) (Goldratt 1992). Although most manufacturing systems usually have one main bottleneck, in all but the simplest applications, bottlenecks are not static, but rather shift between different machines. These shifts may for example be due to the sequence of random events or due to a gradual change in the manufacturing system. A non-bottleneck machine may become a bottleneck, for example due to a machine failure, and similarly a bottleneck machine may become a non-bottleneck machine. Over longer periods of time, a system therefore may not only have one primary bottleneck, but also secondary and tertiary bottlenecks, i.e. machines which are also occasional bottlenecks, yet to a lesser extent than the primary bottleneck. This

1
2
3 highlights the fact that it may be possible to improve throughput by not just focusing
4 on the primary bottleneck. Roser (Roser 2002) deals with shifting bottleneck
5 detection and uses an example with a job-shop arrangement with alternative routing,
6 and alternative resources.
7
8
9

10
11
12 In Muthiah et al's paper (Muthiah 2007) the authors define a metric known as
13 Overall Throughput Effectiveness (OTE) which builds upon the well established
14 Overall Equipment Effectiveness (OEE), which is widely accepted as a primary
15 performance metrics in manufacturing (Hansen 2002). While OTE has the benefits of
16 monitoring total factory performance, being a useful aid in machine bottleneck
17 detection in a varied selection of workflow sub-systems (e.g. machines in parallel,
18 multiple routing) workers and skills were not considered during the development of
19 the metric. Li et al. (Li 2009b) use analytical methods for bottleneck detection, based
20 on historical data on machine blockages and starvations and also take into account
21 machine failure times. The authors also note that using analytical methods for
22 bottleneck detection are generally only appropriate for relatively simple workflows
23 comprised of machines in sequence. However in Li's other paper (Li 2009a) an
24 analytical data-driven method is used for bottleneck detection in a complex
25 manufacturing system. Again however, workers are not dealt with in these two of Li
26 et al's papers. For larger more complex systems, that include worker resources
27 simulation would need to be used.
28
29
30
31
32
33
34
35
36
37
38
39
40
41

42 All of the aforementioned approaches for bottleneck detection generally
43 assume heavily automated flow-shop production times where the human element can
44 be ignored. However, in a large proportion of enterprises, workers are an important
45 resource and cannot be disregarded when looking at bottleneck detection. The reason
46 they tend to be ignored in simulation, is that they add an extra layer of complexity and
47 they also necessitate the modelling of calendars, shifts, and breaks. Once workers are
48 included in bottleneck analysis, it muddies the water and the conventional analyses
49 may not suffice. Consider a simple production line with four machines and a single
50 worker. Each machine has the same processing time and requires the worker to
51 operate it. In this case the bottleneck resource is not any of the machines, but is in
52 fact the worker. If we were to add a second worker, we will double the throughput.
53 Another important issue which is not dealt with are machine modes of operation, such
54
55
56
57
58
59
60

1
2
3 as setup, handling, and maintenance. Each mode may require a different set of skills
4 to operate e.g. a technician may be required to perform maintenance on the machine,
5 or a fitter and an electrician to setup the machine to process a new product type.
6
7
8
9

10
11 In our research, we have developed a database driven simulation model that is
12 applicable to a generic manufacturing enterprise. All the necessary information about
13 the enterprise such as resources, orders, workflows, and work calendars, is stored in a
14 relational database (RDB), and a simulation engine, eM-Plant, instantiates a model of
15 the enterprise from the database. The workflows we have developed are based on
16 work by Russell et al. (Russell 2004) and Lin et al. (Lin 2009). Our workflows can
17 deal with alternative routings such as OR-Splits and AND-Splits and the
18 corresponding Joins, and also deals with alternative ranked resources. In addition the
19 workflows can also contain subcontracting activities. On the resource level, machines
20 can have multiple modes of operation such as setup and preventive maintenance,
21 sequence dependent setup times, and workers can possess multiple skills, with an
22 efficiency level being defined for each worker skill combination.
23
24
25
26
27
28
29
30
31
32

33
34 For the rest of this paper, in section 2 we define the metrics which we use to
35 measure the contributing effect each resource has on the overall system
36 constrainedness, and in section 3 we present some computational examples for
37 calculating RCM values. Section 4 covers the determination of RCM values through
38 a simulation of a real-life production plant.
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

2. Resource Constraint Metrics

2.1 Modelling Workers, Skills, and Efficiencies

Firstly we define a machine as an internal locus where a manufacturing activity is performed. A machine can process orders as well as preventive maintenance requests (PMRs), and emergency maintenance requests (EMRs). A skill meanwhile, is a worker ability that is requested by a machine in its particular mode of operation. For example a lathe may require a skill 'Can operate Lathe' in its run mode of operation, and the skill 'Can repair Lathe' in its emergency maintenance mode of operation. A worker can have a set of skills, and each skill for a particular worker can have an efficiency between 0% and 100%. A skill can be regarded as an abstract resource, and a worker can only provide one of their skills at a time. The efficiency increases the processing time for a machine as follows:

$$P_a = \frac{P_n}{\eta}$$

P_a = Actual Processing Time

P_n = Nominal Machine Processing Time

η = Efficiency of Worker or Average Efficiency of Team of Workers ($0 < \eta \leq 1$)

If the nominal processing time for a machine is 10 minutes, and if a worker with an efficiency of 75% is assigned to the machine, then the actual processing time will be 13.3 minutes. We must also define the concept of skill starvation. We say a machine is neglectfully starved for skill when capable workers are busy elsewhere on the shop-floor, and therefore unavailable to satisfy a request from the machine. Orders and PMRs may also be in queue for a machine that is suffering from absenteeism induced skill starvation, which means that there are no workers on the shop-floor for the current shift capable of operating the machine in either its standard modes of operation (run, setup, load, unload, teardown) or its relevant preventive maintenance mode. Our definition of starvation is not to be confused with the concept of a machine being starved for parts. In traditional simulation analysis, a machine that is idle and starved for parts is often frowned upon, and it is usually pointed out that its utilisation should be increased. However, since our simulator, models the entire enterprise, all parts/orders can be traced back to external customer demand. A machine that is idle is not necessarily a negative thing; if there is no

associated customer demand there may be no point providing the machine with extra parts to produce (the exception being if the enterprise is building to stock). However, a machine that is starved for skill is impeding the movement of parts through the system, parts which will generally have actual associated customer demand.

2.2 Formulae for Resource Constraint Metrics

For each machine and skill the simulation model will calculate a value known as the Resource Constraint Metric (RCM) to quantify how much of a limiting factor that resource is to the manufacturing operations, and thus help identify it as a resource bottleneck. The higher the RCM value, the more of a limiting factor that resource is. Shown below are the actual formulae for calculating the RCMs for machines and skills.

Machine RCM

$$\{A_i\}_{i=1}^M = \sum_{j=1}^N [(Q(p)_{ij} - Q(p(e))_{ij})]$$

Skill RCM

$$\{A_k\}_{k=1}^S = \sum_{i=1}^M \sum_{j=1}^N [(Q(s)_{ijk} + Q(p(e))_{ijk}) + H_{ijk}]$$

M	=	Total number of Machines
S	=	Total number of Skills
N	=	Total number of Orders, PMRs, and EMRs
i	=	Index for Machines (M_i)
j	=	Index for Orders, PMRs, and EMRs (R_j)
k	=	Index for Skills (S_k)
$Q(p)_{ij}$	=	Queue time for R_j at M_i when M_i was processing
$Q(p(e))_{ij}$	=	Queue time for R_j at M_i when M_i was excess processing ¹
$Q(p(e))_{ijk}$	=	Queue time for R_j at M_i when M_i was excess processing with S_k
$Q(s)_{ijk}$	=	Queue time ² for R_j at M_i when M_i was starved for S_k

¹ Excess processing time that is incurred due to worker inefficiencies. See first paragraph on next page for a more detailed explanation.

² EMRs will never be in queue for a machine. When a machine fails, an EMR is generated and instantly replaces the order that was on the machine at the time of failure.

H_{ijk}	=	Time spent by R_j on M_i when M_i was starved for S_k
A_i	=	Resource Constraint Metric for M_i
A_k	=	Resource Constraint Metric for S_k

2.3 Explanation of Resource Constraint Metric Formulae

Excess processing time is incurred due to worker inefficiency. This will only be relevant for machines that have workers assigned to them, and those workers have efficiency levels of less than 100% for a particular skill. For example, consider the nominal run-time for a machine is 10 minutes. If a worker with 80% efficiency for the requested skill by the machine is operating the machine, then the actual run-time will be 12.5 minutes. Other orders in queue will be waiting and thus contribute to the RCM, but for each order we add only 10 minutes for the RCM for the machine and add the other 2.5 minutes to the RCM for the skill. The 2.5 minutes is the excess processing.

We also separate out the different RCMs based on the shift when they were recorded. Separating out RCMs based on shift is useful in determining not only where the problem skill is, but when it is. For example, the main contributing factor to a high RCM for a particular skill could be due to a higher absenteeism rate during the night shift than the day shift. Sample RCMs are given in (Figure 1). We can see that we have a higher RCM value for skill 2 during the night shift than the day shift. Therefore, one possible recommendation to reduce the SCM is to increase the skill-set of a worker who works the night shift, by providing them with skill 2.

It should be pointed out that the equation on the previous page for calculating skill RCMs is applicable to multiple skill requests. For example a machine is neglectfully starved for skill for 10 minutes and is requesting 2 workers with skill A and the only two workers who possess this skill are on their lunch-break. If the machine has been assigned an order, and there are no other orders or PMRs in queue for the machine, then a value of 20 minutes will be appended to the RCM for that machine. If one worker had been idle and the other on their lunch-break, then the recorded value would be only 10 minutes.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

Another thing to note is that even though we model machine alternatives, an order can only ever be in queue for a single machine at any instant in time. When an order moves onto its next activity, and there are multiple machine alternatives, the order is moved to the queue which will result in it being processed fastest³. A simple analogy is a person in a supermarket; they look at their options in terms of check-outs, and they will choose the check-out which they believe will result in them leaving the supermarket quickest. It is possible for an order to switch to a queue for a different machine however. For example if a machine fails suddenly, the orders that were in queue for it may be moved over to a queue for an alternative machine, if the time to repair the first machine is expected to be longer than setting up the alternative machine.

The calculations do not attempt to weight orders based on priority, quantity, or cost. We do not weight by priority, since higher priority orders should spend proportionally less time in queue for machines, and on machines that are starved for skill than lower priority orders anyway. Also, we do not weight by quantity or cost, since the calculations also include maintenance requests; if we were to try and weight an order, we were have to apply some arbitrary weighting variable to maintenance requests. However, future work may dictate that it is necessary to weight each order by the total euro-cost associated with each order. In this way a high cost order will contribute proportionally more to the RCM for the constrained resource, than lower cost orders. The weighting variable for maintenance requests would be a euro-cost that captures the delay of a PMR in queue for a machine, or a starved machine that is processing a PMR or EMR. Assigning a weighting variable to maintenance requests is in the realm of fuzzy logic however; intuitively we can say that the RCM from a machine that fails unexpectedly and is waiting for a worker rated to repair it, should be worse than the RCM for a machine that is starved for skill while processing a low priority work order.

When we run a simulation each machine and skill will have a generated RCM value, which will generally have units of hours. We are only interested in relative

³ This is a prediction within the simulation at run-time however, and may actually turn out to be wrong. For example if the selected machine fails unexpectedly, the alternative machine may have been a better choice.

1
2
3 RCM values, and the resource with the highest RCM value will generally be the one
4 we should focus on if we want to reduce the system impedance. At this stage we also
5 define the concept of the System/Shop-Floor Constraint Metric (SCM) which is
6 simply the sum of all the individual RCMs. We may find that if we reduce the RCM
7 of one resource (e.g. reducing the processing time of a machine) then the RCM of
8 another resource may increase. In this situation we could find that the SCM remains
9 the same, indicating that there is no overall net improvement of the system. This will
10 be the case if in a simple production line we reduce the processing time of a non-
11 bottleneck machine.
12
13
14
15
16
17
18
19
20

21 *2.2 Advantages of Resource Constraint Metrics*

22 The advantage of using RCMs over simply trying to locate the bottleneck machine, is
23 that in any manufacturing systems which are other than the most simple cases, we
24 need a way of ranking the contribution of each resource to the system
25 constrainedness. The case of performing analysis on a manufacturing system and
26 claiming that 'machine X is the bottleneck' will generally only be tenable for a single
27 sequential production line, with no workers modelled. Roser's paper (Roser 2002)
28 has expanded the research by creating a methodology to identifying secondary and
29 tertiary bottlenecks, but the RCM is a more flexible way of quantifying (relatively)
30 the contribution of each machine to the overall system constrainedness. Once we
31 factor in workers and skills, which are often neglected in simulation, but a critical
32 resource in most enterprises, we need a way to differentiate between the contributing
33 effect of machines and workers. The RCMs do this in a intuitive manner, and can be
34 used to aid in real-world decision-making. By looking at the status quo or as-is
35 situation on the manufacturing shop-floor, the RCM values will inform the user where
36 best to focus their efforts in order to reduce the overall system constrainedness e.g.
37 add a new machine, reduce the processing time of an existing machine, add a new
38 worker, or upskill an existing worker. This is particularly relevant for small
39 enterprises with relatively small workforces, and where workers must be included as a
40 resource in the simulation of the system; upskilling an existing worker could
41 potentially have an equivalent rate-of-return as purchasing a new machine but without
42 the capital investment.
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

1
2
3
4 Subsequent simulations with the changes in place should support the RCM
5 data i.e. we should see a reduction in SCM, and we may see a reduction in the number
6 of late satisfied customer demands (a positive consequence of orders flowing through
7 the system quicker). We may also see an increase in the finished goods inventory,
8 since a decrease in SCM may result in reduced work order fulfilment lead time. It is
9 important to note however, that a high RCM for a skill associated with a particular
10 machine can be tackled by reducing the processing time for the machine. This is
11 because by reducing the processing time, there is less opportunity and time for the
12 machine to be starved for skill.
13
14
15
16
17
18
19
20

21 Another advantage of using these metrics is that the coding for the calculations
22 is incorporated directly into a database driven general simulation model that has been
23 developed in eM-Plant. Our approach instantiates a model of an enterprise from a
24 RDB which results in significantly reduced build effort. Since the coding for
25 calculating RCM is build into the simulation model in eM-Plant the coding for the
26 metrics are not specific to a single case study. Multiple instances of enterprises can
27 be represented in relational database format, and the RCM values calculated, once the
28 model is automatically instantiated in eM-Plant. The common approach to building
29 simulation models is to build them ad hoc; from the ground up, using components and
30 elements in a simulation application user interface. For example the paper by Alfieri
31 (Alfieri 2009) describes a relatively complex simulation of a cardboard company
32 which contains elements such as multi-machine stations, sequence-dependent setup
33 times, resource calendars, and external operations. Similarly the paper by Nacsa
34 (Nacsa 2008) demonstrates a model that includes components such as machines,
35 shifts, and workers. These models are however each specific to the shop-floor of a
36 single manufacturing firm. In the case of the paper by Nacsa, the model was built
37 using Siman, a simulation application.
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52

53 Our simulator is capable of instantiating models from a database at least as
54 complex as described in (Alfieri 2009) and (Nasca 2008) and will produce RCMs, and
55 other decision oriented KPIs. The RCMs could therefore potentially be used for
56 controlling resources on an operational level, in a similar manner to the paper by Li et
57 al. (Li 2009c) where the authors have achieved real time production improvement
58 through bottleneck control.
59
60

3. Computational Examples

3.1 Example One

This section will describe three computational examples. The first example deals with a simple sequential workflow with four machines each with a constant processing time of one minute. We start the simulation with 200 orders in queue for the first machine, and we end the simulation when all the orders have left the system. At the end of the simulation we find the following RCM values as shown in (Table 1).

As expected, the only non-zero RCM value for the machines is for the first machine. Since the first machine feeds machine B at a rate of one order per minute, and machines B, C, D have a processing time of one minute, no orders will be in queue downstream of machine A. Also, in this simple example we are not dealing with workers, hence none of the machines will undergo skill starvation. The reason machine A has a non-zero RCM value is because at the start of the simulation there are 200 orders in queue for the machine. From the formula for RCM for machines, the RCM value for a machine is proportional to the queue size or WIP (work-in-process) for the machine; the more parts in queue for a machine, the higher the RCM for the machine, and also for the skill associated with the machine if the machine undergoes skill starvation. The RCM for example one can be calculated by noting that at time 1, there will be 199 orders that have been in queue for machine 1 for 1 minute. Hence we add 199 minutes to the RCM for machine 1. At time 2, there will be 198 orders that have been in queue for machine 1 for 1 minute, so we add on 198 minutes to the RCM value. Therefore, the total RCM value at the end of the simulation should be given by:

$$\sum_{i=1}^n i = \frac{n^2 + n}{2}$$

Which for a value of $n = 199$ yields an RCM value of 19,900. The reason for the discrepancy between the calculated value of 19,900 and the simulated value of

1
2
3
4 20,225 is due to a condition known as event time shifting in the simulation model,
5 whereby events which are technically supposed to occur at the same time they were
6 triggered, must be scheduled for 1 second in the future. Events must be dealt with in
7 this manner in eM-Plant in order to avoid events being inadvertently bypassed by the
8 simulator. However, the error in RCM values for this simple model is only
9 approximately 1.6%, and since the simulator is aimed at simulating enterprises over
10 relatively long periods of time, this incurred error is negligible, and should not
11 negatively affect the output results in a decision making context. Next, if we reduce
12 the processing time of machine A to 30 seconds (New Scenario 1), we obtain the
13 following results as shown in (Table 2).
14
15
16
17
18
19
20
21
22

23 Because machine A can process orders twice as fast as before, a queue builds
24 up in front of machine B. Hence machine B will now have a non-zero RCM. As
25 expected however, the SCM stays the same at 20,225 minutes. This indicates that
26 halving the processing time of machine B has had no effect on the overall system
27 constraint metric. Only if we reduce the processing time of all four machines will we
28 notice a reduction in the SCM. Shown in (Table 3) are the results when the
29 processing time of all the machines is 30 seconds (New Scenario 2). The value of
30 10,278 minutes is approximately half of 20,225 minutes; we have reduced the SCM
31 by 50% which corresponds to a doubling of throughput.
32
33
34
35
36
37
38
39
40

41 **3.2 Example Two**

42 In the second example, we use the model from first example with some alternations.
43 The processing times of all machines are now stochastic, uniformly distributed
44 between 55 and 60 seconds, and orders are now released to the system at a rate of one
45 order per minute. This will have the effect of reducing the queue in front of the first
46 machine. Since the processing times are stochastic we run the simulation for 10
47 replications until all the orders have left the system. Shown in (Figure 2) are the
48 RCM values obtained for this example.
49
50
51
52
53
54
55

56 For each machine, from the X-axis to the thick black line in the middle of the
57 grey box, the corresponding Y value is the mean RCM. From this value, to the top
58
59
60

1
2
3 and bottom of the grey box, represents one standard deviation⁴ above and below the
4
5 mean. We see that machine A has a zero RCM value. Since orders are now released
6
7 at a rate of one per minute, and the longest processing time capable for the machine is
8
9 one minute, there will never be orders in queue for machine A. The RCM values
10
11 increase the further downstream we go. This is due to the cumulative effect of the
12
13 variability of the processing times. Next if we change the processing time of machine
14
15 C to a uniform distribution of between 80 and 85 seconds we obtain the following
16
17 results for the RCMs, for 10 replications as shown in (Figure 3).

18
19 We see that machine C dominates. Again this is to be expected since it is now
20
21 the bottleneck because it has the longest processing time of the four machines, and
22
23 thus will have the largest queue in front of it. Machine B has a non-zero RCM but it
24
25 is dwarfed by the RCM for machine C. The RCM value for machine D is now zero,
26
27 since machine C will release orders at a rate of less than one per minute, and since the
28
29 processing time of machine D will never exceed one minute, there will never be any
30
31 orders in queue for machine D. If we want to make improvements to the system, we
32
33 should concentrate on machine C. Also it is worth noting that the RCM value for
34
35 machine D in (Figure 3) is 130 hours, whereas when all the machines had the same
36
37 processing time, the highest RCM value for machine D is around 0.015 hours (54
38
39 seconds) (Figure 2). There is a significant difference in scale.

40 41 **3.3 Example Three**

42
43 In the third example we introduce workers and skills, job dependent processing times,
44
45 and multiple workflows. We have three sequential workflows in total

- 46
47
- 48 • Workflow 1 = {Activity A → Activity B → Activity C → Activity D}
 - 49 • Workflow 2 = {Activity B → Activity A → Activity D → Activity C}
 - 50 • Workflow 3 = {Activity C → Activity B → Activity A → Activity D}

51
52 Activities are performed at machines as follows:

- 53
54
- 55 • Activity A performed at {Machine A}
 - 56 • Activity B performed at {Machine B}
 - 57 • Activity C performed at {Machine C}

58
59
60 ⁴ We use sample standard deviation, with Bessel's correction for measuring the spread of the results, which we store in a separate RDB. We do not use confidence intervals, since confidence intervals assume that the outputted results follow the normal model distribution which may not be the case.

- Activity D performed at {Machine D}

We assume machines only have a single mode of operation “Run” and machines request the following skills for their “Run” mode of operation. Also each machine sorts jobs in the queue by their priority, and then by shortest processing time.

- Machine A : Run requests skills {Skill A}
- Machine B : Run requests skills {Skill B}
- Machine C : Run requests skills {Skill C}
- Machine D : Run requests skills {Skill D}

We have three workers with the following skill-set:

- Andrew has skills {Skill A}
- Brian has skills {Skill B}
- Con has skills {Skill C, Skill D}

Processing times are dependent on job ID and we have 6 jobs in total. The processing times have units of minutes and are given in (Table 4). The information on jobs is shown in (Table 5). Each job is given the same arbitrary scheduled release date, which corresponds to time zero in the simulation model, and the same internal due date. The priority of an order dictates its queuing priority and worker dispatching priority i.e. a high priority order will be given higher preference to be processed ahead of a lower priority order.

An overview of the relationships between workflows and skills is given for workflow 1 in (Figure 4). The linkage type between activities is ‘sequence’, and simply defines a precedence constraint dependency between activities i.e. we can not start activity B until activity A is completed. Looking at this diagram, we see that Con will need to share his time between machines C and D. Therefore there is a potential for these two machines to be starved for skill.

We run the simulation for a simulated time of 45 minutes, after which all the jobs have been processed. Shown in (Figure 5) is a Gantt chart for the machines. We use a different colour to denote each job, and when a job is on a machine that is starved for skill, we use a thin binary colour strip with the same colour as the job plus the colour red. This Gantt chart does not try to represent jobs which are queued for machines. For example, job 5 (yellow) between time 20min and 25min is in queue

1
2
3
4 for machine D, but during this time period machine D is busy processing job 4 (grey);
5 job 5 is not shown in the Gantt chart for this period. Also, machine C starts
6 processing job 4 ahead of job 6 because job 4 is higher priority than job 6. Machine
7 A processes job 1, job 5, and job 3 in that order because they have increasing
8 processing times. If we look at machine C at time 14 we see that job 5 is loaded onto
9 the machine, but the only worker capable of operating machine C is Con and he is
10 currently busy with job 2 on machine D. Therefore machine C is neglectfully starved
11 for skill C for two minutes. When Con finishes job 2 he moves up to machine C and
12 begins processing job 5. Also, as expected at any instant in time, machines C and D
13 will never be simultaneously processing. The worker Andrew only possess skill A,
14 and since machine A is the only machine that requests this skill, Andrew is confined
15 to machine A. This implies that it is impossible, under the current conditions, for
16 machine A to be starved for skill. It is the same situation with the worker Brian and
17 machine B. Shown in (Figure 6) is a Gantt chart for the jobs. We use a different
18 colour to denote each machine, and when a machine is starved for skill, we use a thin
19 binary colour strip with the same colour as the machine plus the colour red.
20
21
22
23
24
25
26
27
28
29
30
31
32

33
34 We will now show how we calculate the RCM values for machine D and skill
35 D which are values of 17 minutes and 10 minutes respectively. In (Table 6) the
36 second column Q_D contains, for each job, the total time the job spent in queue for
37 machine D. The third and fourth columns break this time down, for each job, into the
38 time when machine D was processing, and when machine D was starved for skill
39 (skill D in this case). For example, looking at (Figure 6) for job 6 we see that it spent
40 a total of 11 minutes in queue for machine D. However, for 3 of these 11 minutes
41 machine D was starved for skill D when processing job 5. So this 11 minutes is
42 separated out to 8 minute for $Q(p)_D$ and 3 minutes for $Q(s)_{DD}$. In the fifth and final
43 column, we record the time for each job, when the job was on machine D and the
44 machine was starved for skill D.
45
46
47
48
49
50
51
52
53

54
55 Q_D = Total Time job in queue for M_D
56 $Q(p)_D$ = Time job in queue for M_D when M_D is processing
57 $Q(s)_{DD}$ = Time job in queue for M_D when M_D is starved for S_D
58 H_{DD} = Time when M_D was neglectfully starved for S_D
59
60

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

When we sum up the third column in (Table 6) we get a value of 17 minutes which is the RCM value for machine D. When we sum up the fourth and fifth columns we get 3 minutes and 7 minutes respectively, and when we add these together we get a value of 10 minutes which is the RCM for skill D. We do not have to consider the other machines in the calculation of the RCM for skill D since they will never be starved for this skill.

dShown in (Figure 7) are the RCMs for all the machines and skills. Looking at this graph, some recommendations could be giving skill C or skill D to the workers Andrew or Brian, or reducing the processing time of machine B. These changes could reduce the SCM and also reduce the makespan of the jobs i.e. the time when the last job leaves the system. However, it should be pointed out that our model here is relatively simple, and over a very short period of time. Decisions made based on RCM values should be for real world systems that are in steady state and over relatively long periods of time. In this way we can be confident in the relativity of RCM values and the dominance of the constraining power of one resource over another on the shop-floor.

4. Real Life Example

In this section we use the simulation model to determine the RCM values for an entire factory. The factory under examination is a SchuF Fetterolf production plant in Ireland that designs and manufactures industrial valves that control, isolate, divert, or sample gases, liquids, slurries, or powders. The main characteristics of the plant being simulated are as follows:

- Simulation time of one year
- 1,600 Work Order / Purchase Order Lines
- 470 Sales Order Lines
- 1,700 Precedence Constraints amongst Orders
- 2,500 Products / Components
- 24 Workers
- 11 Skills
- 18 Machines
- 8 Vendors / Subcontractors
- 150 Workflows

- One 8 hour shift, five day working week
- Multiple Routing, Alternative Resources
- Multiple Machine Modes of Operation
- 1,000 Different Processing Times associated with different workflows and different machines

This information was stored in the companies ERP (Enterprise Requirements Planning) system and was translated and stored in the Relational Database for the simulation model. The primary source of variability being modelled is worker absenteeism and tardiness. Based on the following papers (Barmby 1991), (Bytestart 2008), (Irish Times 2008) we assume the following parameters:

- Probability of being absent is 3.3%
- Probability of being late is 4.4% (However, 65% probability that when a worker is late this will occur on a Monday)
- Average Lateness is 10 minutes, with an upper limit of 18 minutes. Hence, assume actual lateness time is log-normally distributed with mean 10 minutes and standard deviation 2 minutes, where probability of being more than 18 minutes late is less than 5%.

The main caveat with modelling absenteeism and tardiness in this manner is that we assume all workers have an equal probability of missing a day of work, or being late for work. In reality of course, some workers will be more prone to missing work than others. A more accurate way to model this variability is to look at historical data for each worker and assign a probability for each worker. However, this data was not currently available at the time this paper was written. It should be noted that with a probability of being absent of 3.3% and with 24 workers the chances of at least one worker being absent on a given day is greater than 55%.

Another source of variability in the model is that equally ranked alternative resources are randomly selected. For example, if a machine requires a worker with the skill 'Can Setup' and two equally ranked workers possess this skill, then one of the workers will be selected at random. The same rule applies to alternative machines for a particular activity. We run the simulation for ten replications, once with worker

1
2
3 absenteeism and worker tardiness being modelled, and for another ten replications
4 with absenteeism and tardiness switched off.
5
6
7

8
9 We test for a statistically significant difference (with a significance level of 5%)
10 between the mean total skill RCM for the two different experiments using R, a
11 statistical analysis software package. Firstly we perform a Kolmogorov-Smirnov test
12 on both sets of data to see if there is sufficient evidence that the data deviates from the
13 normal model. We get p-values greater than 5% meaning we can assume the data is
14 normally distributed. Next we check the data sets to see if we can assume equal
15 variances. In this case we get a p-value less than 5% indicating that the variances are
16 unequal. This result was to be expected since the lions-share of variance is from
17 modelling absenteeism and this is only present in the first experiment. The only
18 source of variance present in the second experiment is from the random selection of
19 equally ranked alternative resources. Finally we perform a Welch test for testing for
20 significance difference between means with unequal variances and we get a p-value
21 less than 5% indicating that the means are not equal. This is evident from Figure 8.
22 This means that when we eliminate worker absenteeism and tardiness this translates to
23 a 79% reduction in skill RCM.
24
25
26
27
28
29
30
31
32
33
34
35

36
37 When we perform a similar statistical analysis on the total machine RCM we find
38 that there is no evidence of a significance difference in means of total machine RCM
39 between the two experiments, and this can be seen in Figure 9. This result was to be
40 expected since eliminating worker absenteeism and tardiness should not significantly
41 reduce the machine RCMs. If instead the experiments were only looking at the effect
42 of machine break-downs, which can be regarded as the machine equivalent of worker
43 absenteeism, then we should find a significant difference between machine RCMs and
44 a possible significant difference between skill RCMs. Machine RCMs and skill
45 RCMs in this context would be dependent on the mean time to repair the machine
46 (MTTR), and the availability of skills to repair the machine.
47
48
49
50
51
52
53
54
55

56
57 When we look at the System Constraint Metric (SCM) which is simply the sum of
58 skill RCMs and machine RCMs we see that machines are the main contributor to
59 SCM (See Figure 10.). The final result is that based on our simulation model of the
60 SchuF plant, when we eliminate worker absenteeism and tardiness the SCM reduces

1
2
3 significantly by an average of 4.1%. Further work would involve analysing historical
4 data and obtaining absenteeism and tardiness percentages based on each worker.
5
6 Analysis of the different skill RCMs could also inform us which skills are critical, and
7
8 simulation could also inform us the effect of absenteeism from different workers on
9
10 the system constrainedness.
11

12 13 14 **5. Summary**

15
16 The metrics presented in the paper are an intuitive and highly flexible way of
17
18 determining the constraining power of a resource in a manufacturing system, via
19
20 simulation. They are superior to traditional bottle neck detection methods by taking
21
22 into account, many aspects of a manufacturing that are usually ignored in simulation
23
24 models, such as workers, calendars, multiple modes of machine operation, and
25
26 complex workflows. Resource constrains metrics are a way of quantifying the
27
28 constraining power of worker-skills as well as machines, and can aid in real-world
29
30 decision making, by determining which resource should be focussed on and
31
32 improved, in order to reduce the overall system constrainedness. This will have the
33
34 effect of increasing throughput or at least providing the capacity for increased
35
36 throughput. Additionally it can result in improved customer demand satisfaction,
37
38 since reduced system impedance can mean a commensurate reduction in the number
39
40 of incidents where a customer's sales order is satisfied late. Further work using these
41
42 metrics and one that the authors are currently working on is using these metrics to
43
44 identify the most constrained skill, and simulating worker trainer, and how this
45
46 improves the economic performance of the organisation e.g. upskilling worker
47
48 Andrew with Skill B with yield the best economic performance compared with all
49
50 other worker-skill training combinations.
51
52
53
54
55
56
57
58
59
60

References

- Alfieri, A. (2009). "Workload simulation and optimisation in multi-criteria hybrid flowshop scheduling: a case study." International Journal of Production Research Vol. 47, 18, 5129-5145.
- Barmby, T., Orme, C., Treble, J. (1991). "Worker Absenteeism: An Analysis Using Microdata." The Economic Journal Vol. 101, 405, 214-229
- Blackstone, J. H. (2001). "Theory of Constraints - A Status Report." International Journal of Production Research Vol. 39, 6, 1053-1080
- Bytestart (2009, December 20). "Worker lateness costs business £66 million every year", bytestart.co.uk - The Small Business Portal
- Cox, J., Spencer, M. (1997). "The Constraints Management Handbook", Boca Raton, Florida, St Lucie, Press
- Goldratt, E. M. (1993). "The Goal: A Process of Ongoing Improvement", New York, North River Press, Inc.
- Hansen, R. C. (2002). "Overall Equipment Effectiveness - A Powerful Production/Maintenance Tool for Increased Profits". New York, NY, Industrial Press.
- Irish Times (2008, September 2) "Absenteeism Issues", IrishTimes.com
- Law, A., Kelton, D. (2000). "Simulation Modeling & Analysis", McGraw Hill.
- Lawrence, S., Buss, A. (1994). "Shifting Production Bottlenecks: Causes, Cures, and Conundrums." Journal of Production and Operations Management Vol. 3, 1, 1059-1478
- Li, L. (2009a). "Bottleneck Detection of complex manufacturing systems using a data-driven method" International Journal of Production Research Vol. 47, 24, 6929-6940.
- Li, L., Chang, Q., Ni, J. (2009b). "Data Driven Bottleneck Detection of Manufacturing Systems." International Journal of Production Research Vol. 47, 20, 5019-5036.
- Li, L., Chang, Q., Ni, J. (2009c). "Real time production improvement through bottleneck control " International Journal of Production Research Vol. 47, 21, 6145-6158.
- Lin, H., Fan, Y., Newman, S. (2009). "Manufacturing Process Analysis with Support of Workflow Modelling and Simulation " International Journal of Production Research Vol. 47, 18, 5129-5145.
- Muthiah, K., Huang, S. (2007). "Overall throughput effectiveness (OTE) metric for factory-level performance monitoring and bottleneck detection." International Journal of Production Research Vol. 45, 20, 4753-4769
- Nacsa, J. (2008). "SIMAN Simulation of Assembly Operations in a Furniture Company."
- Roser, C., Nakano, M., Tanaka, M. (2002). "Shifting Bottleneck Detection". Winter Simulation Conference, Orlando, Florida

Russell, N., Van Der Aalst, W. (2004). "Workflow Resource Patterns: Identification, Representation and Tool Support", Eindhoven University of Technology.

Machine	Resource Constraint Metric / Minutes
Machine A	20225
Machine B	0
Machine C	0
Machine D	0
SYSTEM	20225

Table 1. RCMs for Computational Example 1 (Baseline)

Machine	Resource Constraint Metric / Minutes
Machine A	10278
Machine B	9947
Machine C	0
Machine D	0
SYSTEM	20225

Table 2. RCMs for Computational Example 1 (New Scenario 1)

Machine	Resource Constraint Metric / Minutes
Machine A	10278
Machine B	0
Machine C	0
Machine D	0
SYSTEM	10278

Table 3. RCMs for Computational Example 1 (New Scenario 2)

	Machine A	Machine B	Machine C	Machine D
Job 1	1	4	4	1
Job 2	6	3	3	2
Job 3	5	5	2	3
Job 4	4	5	5	5
Job 5	2	2	4	4
Job 6	3	4	3	4

Table 4. Processing Times (in Minutes) for Computational Example 3

Job ID	Assigned Workflow	Priority	Scheduled Release Date	Internal Due Date	Status
Job 1	Workflow 1	Low	01/01/08	02/01/08	Unreleased

Job 2	Workflow 2	Normal	01/01/08	02/01/08	Unreleased
Job 3	Workflow 1	Low	01/01/08	02/01/08	Unreleased
Job 4	Workflow 3	Normal	01/01/08	02/01/08	Unreleased
Job 5	Workflow 1	Low	01/01/08	02/01/08	Unreleased
Job 6	Workflow 3	Low	01/01/08	02/01/08	Unreleased

Table 5. Jobs for Computational Example 3

Job ID	Q_D	$Q(p)_D$	$Q(s)_{DD}$	H_{DD}
Job 1	0	0	0	0
Job 2	0	0	0	0
Job 3	4	4	0	0
Job 4	0	0	0	2
Job 5	5	5	0	3
Job 6	11	8	3	2
TOTAL	20	17	3	7

Table 6. Breakdown of Queue/Starvation Time for Computational Example 3

	Replication Number									
	1	2	3	4	5	6	7	8	9	10
Skill RCM (Hours)	64	61	37	25	52	39	12	15	15	30
Machine RCM (Hours)	592	592	591	598	591	592	592	598	588	583
SCM (Hours)	656	653	628	623	643	631	604	613	603	613

Table 7. Simulation Results for Production Plant with Absenteeism and Tardiness

	Replication Number									
	1	2	3	4	5	6	7	8	9	10
Skill RCM (Hours)	5	6	8	10	6	7	7	11	7	7
Machine RCM (Hours)	594	595	596	588	600	594	593	587	595	595
SCM (Hours)	599	601	604	599	606	602	601	598	603	602

Table 8. Simulation Results for Production Plant with no Absenteeism and Tardiness

Figure 1. Sample Resource Constraint Metrics (By Shift)

Figure 2. Resource Constraint Metrics for Computational Example 2 (Baseline)

Figure 3. RCMs for Computational Example 2 (New Scenario 1)

Figure 4. Breakdown of Workflow 1 for Computational Example 3

Figure 5. Machine Gantt Chart for Computational Example 3

Figure 6. Job Gantt Chart for Computational Example 3

Figure 7. Resource Constraint Metrics for Computational Example 3

Figure 8. Skill RCMs for Production Plant

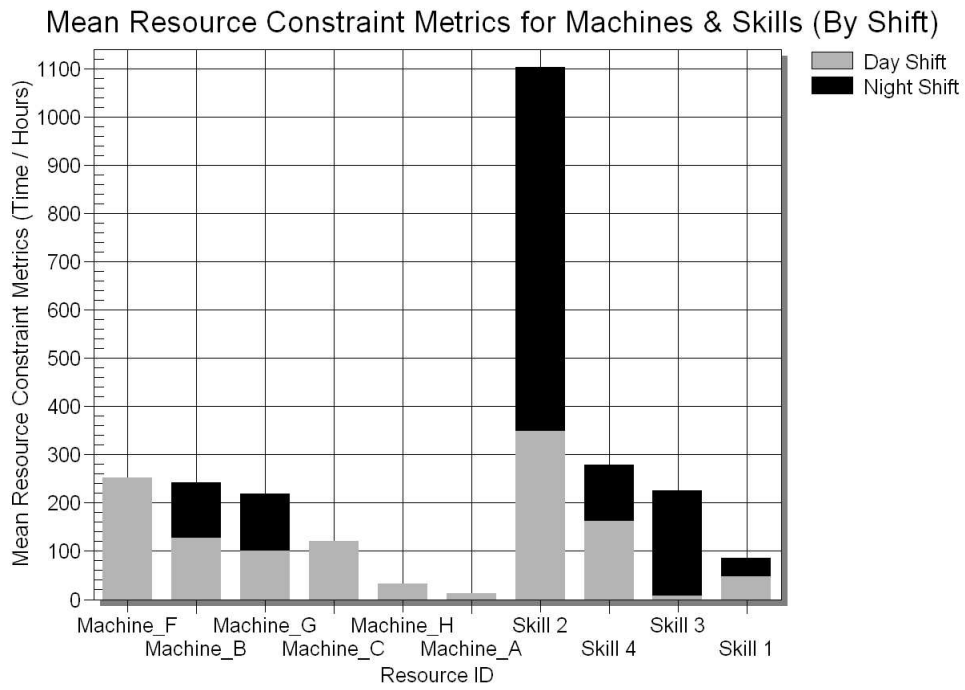
Figure 9. Machine RCMs for Production Plant

Figure 10. SCMs for Production Plant

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

For Peer Review Only

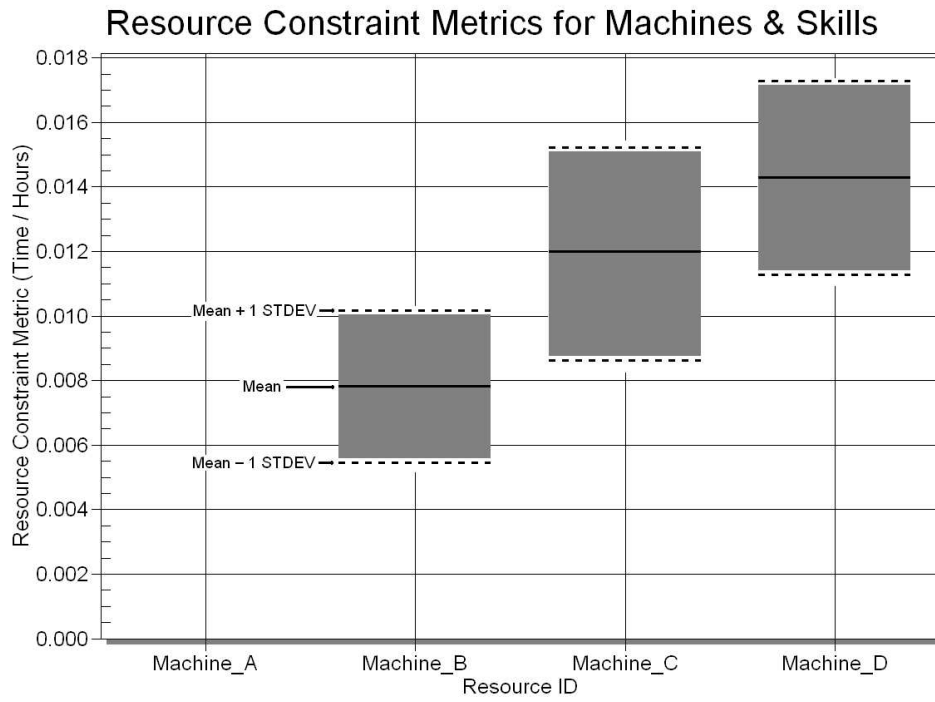
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60



329x235mm (96 x 96 DPI)

view Only

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

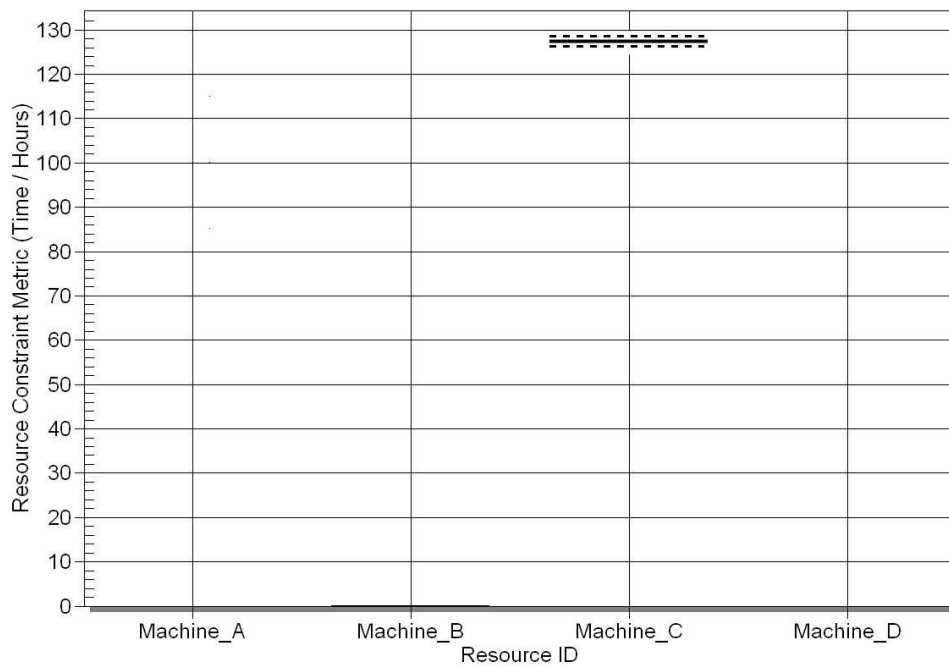


318x232mm (96 x 96 DPI)

View Only

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

Resource Constraint Metrics for Machines & Skills

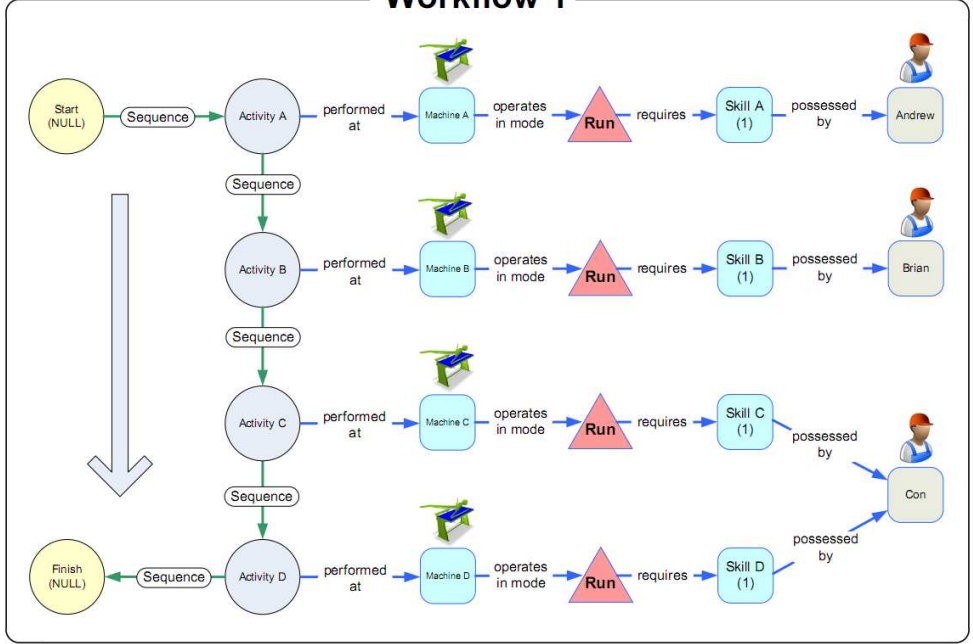


313x232mm (96 x 96 DPI)

Preview Only

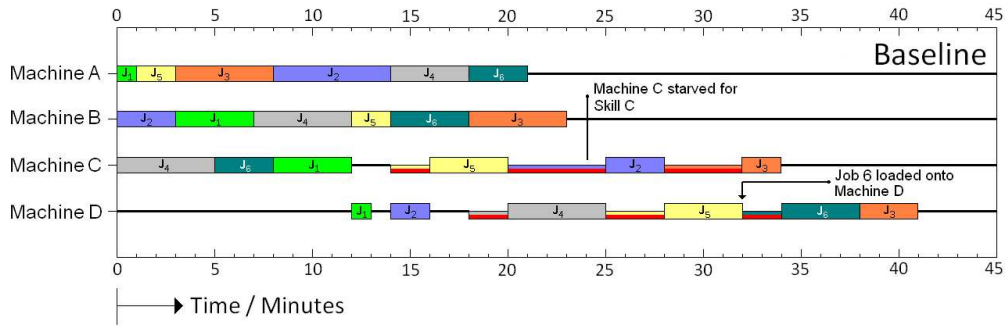
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

Workflow 1



311x211mm (96 x 96 DPI)

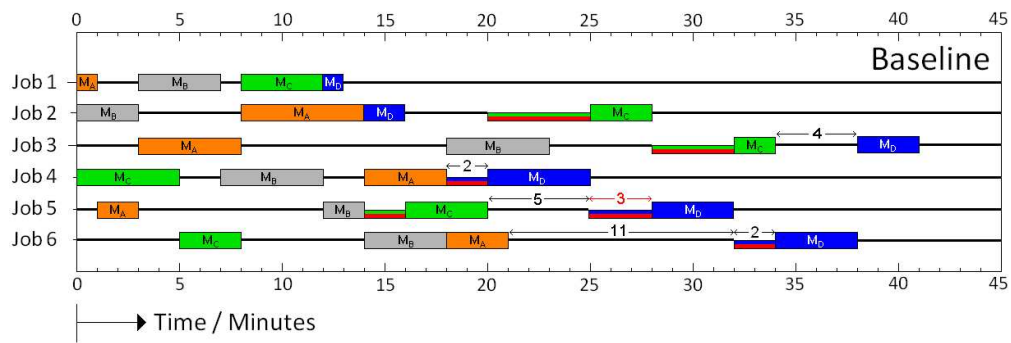
view Only



338x116mm (96 x 96 DPI)

Peer Review Only

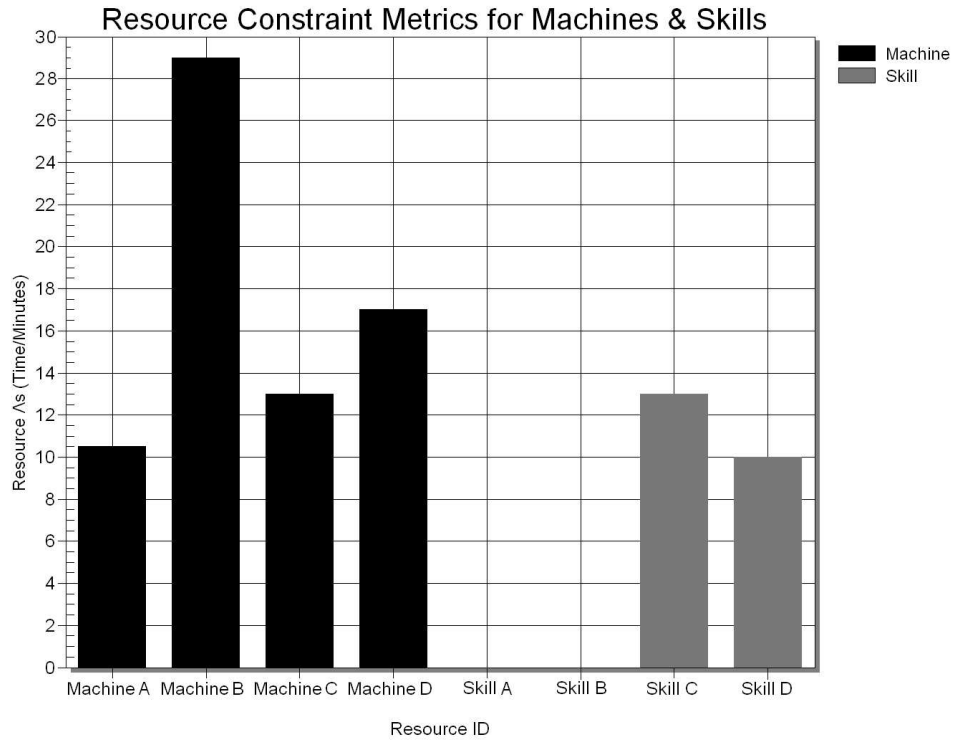
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60



322x115mm (96 x 96 DPI)

Peer Review Only

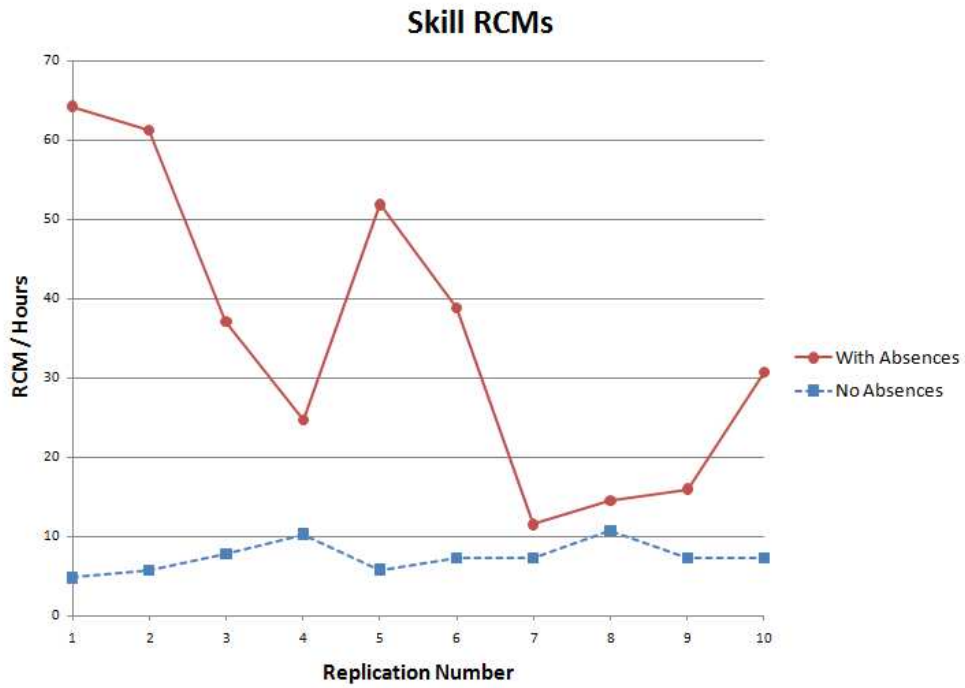
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60



369x283mm (96 x 96 DPI)

Preview Only

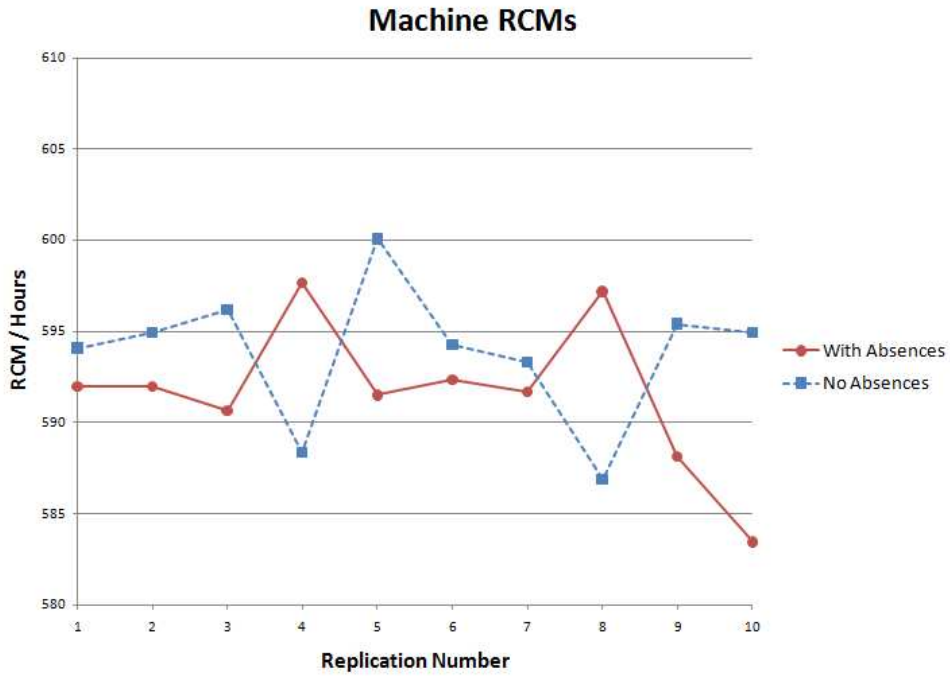
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60



193x139mm (96 x 96 DPI)

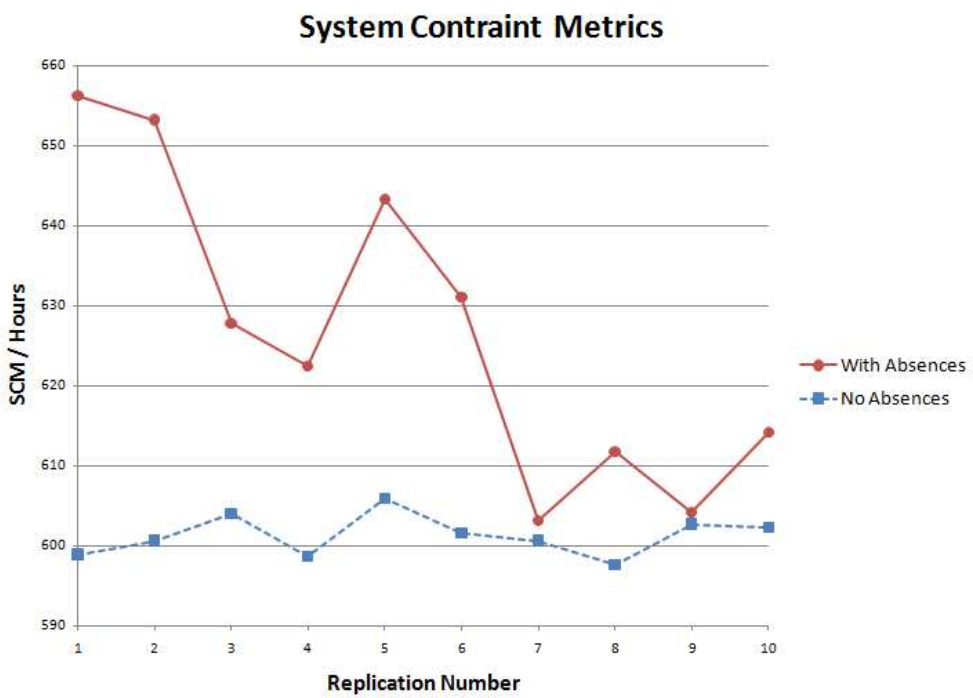
View Only

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60



196x138mm (96 x 96 DPI)

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60



191x138mm (96 x 96 DPI)

View Only