KATHOLIEKE UNIVERSITEIT
LEUVEN

Faculty of Business and Economics

Extending the production dice game

Marc Lambrecht, Stefan Creemers, Robert Boute and Roel Leus

DEPARTMENT OF DECISION SCIENCES AND INFORMATION MANAGEMENT (KBI)

KBI 1005

Extending the Production Dice Game

MARC LAMBRECHT
Research Center for Operations Management, Faculty of Business and Economics, K.U.Leuven, 3000 Leuven, Belgium. E-mail: Marc.Lambrecht@econ.kuleuven.be

STEFAN CREEMERS
Research Center for Operations Management, Faculty of Business and Economics, K.U.Leuven, 3000 Leuven, Belgium. E-mail: Stefan.Creemers@econ.kuleuven.be

ROBERT BOUTE
Operations & Technology Management Center, Vlerick Leuven Gent Management School, 3000 Leuven, Belgium and Research Center for Operations Management, Faculty of Business and Economics, K.U.Leuven, 3000 Leuven, Belgium. E-mail: Robert.Boute@vlerick.be

ROEL LEUS
Research Center for Operations Research & Business Statistics, Faculty of Business and Economics, K.U.Leuven, 3000 Leuven. E-mail: Roel.Leus@econ.kuleuven.be

The production dice game is a powerful learning exercise focusing on the impact of variability and dependency on throughput and work-in-process inventory of flow lines. In this paper we will extend the basic dice game along the following lines. First, we allow that the operations take place concurrently as opposed to the more traditional way of playing the game sequentially. Second, we allow both starvation and blocking of the line. Third, we consider balanced lines with work stations characterized by different degrees of variability. Fourth, we use different sets of dice in order to represent a wide range of coefficients of variation of the production line. The game can be played manually in a classroom setting, but it is also modelled as an easy-to-use simulation tool.

INTRODUCTION

The production dice game is a learning exercise focusing on the impact of variability and dependency on throughput and work-in-process inventory. The game deals with flow-shop layouts, i.e., layouts in which equipment or work stations are arranged according to the progressive steps by which a product is made. An assembly line, in which the path of a product is a straight line, is a good example. Unfortunately, the production rate of a work station can be highly variable due to all sorts of outages (machine failures, repairs, minor stoppages, changeovers, etc…). Variability is inherent in almost every production environment. The work stations are therefore usually buffered with inventory, meaning that work-in-process is stored in front of each station; these buffers can serve to cushion (part of) the variability in the line. The buffer space between two stations is usually limited.

Material moves from one station to the next along the line structure. This creates dependencies, i.e. certain operations cannot begin until other operations have been completed. The combination of dependency and variability creates machine interactions, which take the form of *starvation* and *blocking* and this finally has an impact on the throughput (output per unit time) and the level of work-in-process of the flow-shop. Consider two consecutive machines. If the upstream machine fails to produce, the downstream machine may become starved because its input buffer is empty and therefore it is forced to be idle. If the downstream machine fails, the upstream machine may become inactive because it is blocked due to the limited buffer space between the two machines (the buffer fills up). The frequency of occurrence of starvation and blocking (and consequently the amount of idle time and lost throughput) depends on the size of the buffers. Buffers defer idleness and consequently increase throughput, this of course at the cost of increased inventory.

The dice game illustrates these concepts. It proceeds as follows. Each player in the game represents a work station, so *n* players represent a sequential line of *n* work stations. At each step, the production output of each work station is determined by the outcome of a die roll; in this way, we introduce variability in the output of the station. Inventory buffers are positioned between adjacent work stations. Large buffers will decouple machines and are able to absorb more variability, while smaller buffers will create more starvation and blocking but are less costly in terms of in-process inventory and cycle times.

The dice game can be played manually by means of dice (to decide the machine outputs) and coins or chips (the parts or products that flow through the line), but it can also be implemented as a simulation-based computer application. A typical layout is described in Figure 1. The dice game includes a number of artificial features, but its behaviour conveniently matches that of a real production line. The insights obtained from the game immediately transfer to real-life situations. The game's setup is limited and straightforward, which makes it very attractive to play in a classroom setting. The computerized simulation tool comes in very handy in ensuing debriefing discussions.

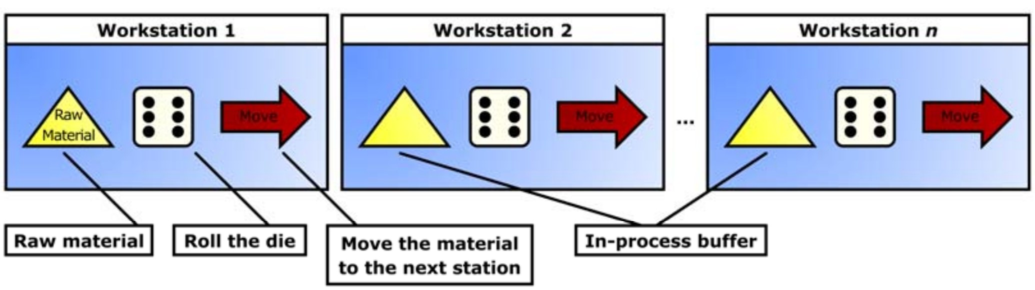

Fig. 1. The basic flow-shop layout of the dice game

LITERATURE REVIEW AND OVERALL GAME SETUP

The dice game has been extensively described in literature, resulting in a wide range of game variants. The game goes back to Goldratt's "boy-scout hike" [1]. Alarcón and Ashley [2] relate the dice game to lean production concepts and use project management as an example. Hilmola [3] introduces system dynamics through the dice game. Umble and Umble [4] and Johnson and Drougas [5] are two papers describing the dice game by means of spreadsheet simulation. In [4] both balanced and unbalanced lines are described, while [5] focuses more on the technicalities of spreadsheet modelling. The last two papers [4] and [5] constitute the basis for the model presented in this article. In the following paragraphs, we outline in which way this paper extends the basic dice game.

First, we allow that the operations take place concurrently as opposed to the more traditional way of playing the game sequentially. In a traditional sequential setup the workstations operate sequentially within a given period, which implies that the processed units of a workstation are available to the subsequent workstation in the same period (the second person rolls the die after the first person has rolled the die). This is not the case when the game is conducted concurrently: in the latter case every workstation operates simultaneously in a given period, independent of the upstream and downstream workstation's production. The production output is then only based on the roll of the die and the surrounding buffers at the beginning of the period. We propose a concurrent variant of the game in this article because this setting better represents paced production lines. Within a given "takt time" (defined in this paper as a period in the game) all work stations perform their work simultaneously and the goods in process move to the next station at the start of a new cycle (takt time). As stated in [5], the steady-state results achieved via either approach (concurrent and sequential) will be comparable (although the probabilities of starvation will be different).

Second, we allow both starvation and blocking of the line. In most papers only starvation is considered. Blocking occurs because the buffers have limited capacity (we impose an upper bound on the in-process inventory at a work station). When the maximum buffer size is reached, the upstream process is blocked. In many realistic automated production settings, the storage areas can hold only a finite amount of material.

Third, we use different sets of dice in order to represent a wide range of coefficients of variation of the production rate ([4] also allows for different levels of variability). This will be explained in the next section, where our model is described in detail.

Fourth, we consider balanced lines (the expected production output is the same for every work station) but we handle work stations with a different level of variability. It is e.g. interesting to examine what the impact is on system performance by having high-variability stations at the start of the production line compared to a high-variability station at the end of the line.

These four extensions turn our dice game into a unique approach in which real-life characteristics are represented more accurately. We do recommend the faculty who are interested in playing the dice game, to be very well informed about the theoretical background of the game, but it goes beyond the scope of this paper to discuss the theoretical insights obtained for buffered serial lines with blocking and starvation. We refer to the following papers for an excellent overview of the theoretical work on the subject: Baker, Powell and Pike [6]; Conway, Maxwell, McClain and Thomas [7]; Dallery and Gershwin [8].

THE SIMULATION TOOL

In this section we describe our simulation tool for the dice game. The tool is available as a stand-alone executable file that requires no specific additional pre-installed software to run. The Dice Game software was implemented in Macromedia Flash using the ActionScript programming language. ActionScript is a cross-platform, object-oriented scripting language that allows the creation of standalone and web-based applications. The game can be played at the following website:
http://www.econ.kuleuven.be/Dicegame

The tool's main interface is represented in Figure 2. If the user presses the question-mark button on the screen, general help will be displayed. Rolling over the button displays screen-specific help.

The serial production line consists of five work stations (represented by circles). The first work station can be considered as a receiving area of raw material (consequently, this station can never be starved), the four other stations are processing units. The last work station serves the customer and can never be blocked. There are two rectangles at the upper left side of the circles: the first rectangle indicates the starting inventory (at the start of the game) and the second contains the maximum number of units allowed in the buffer (the buffer size). These two values are input to a simulation run (and consequently do not change during the game). The square in the middle of the circle reports the result of the roll of a die (the die on the left upper side is the outcome of the die in the previous roll). The outcome of a die roll is the production output of the work station.

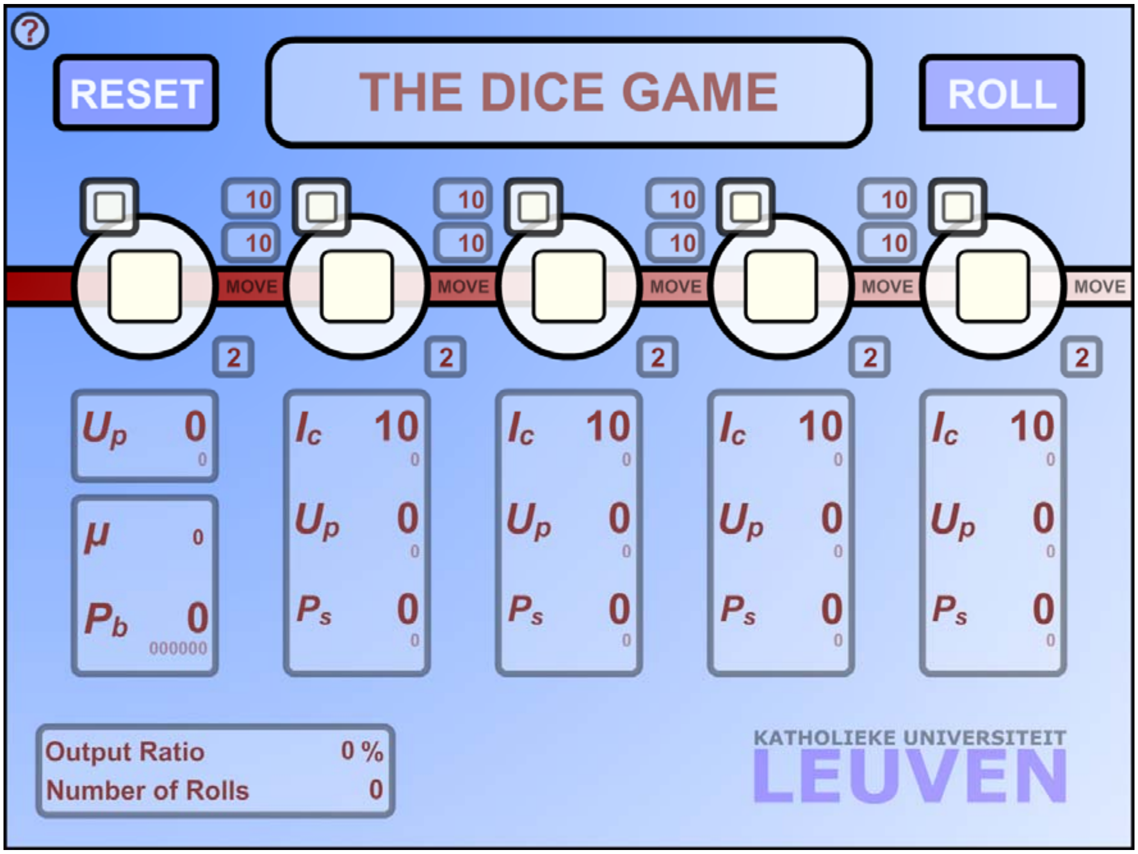

Figure 2 : The program interface

The type of die used is indicated in the box at the bottom right side of the circle. We propose six types of dice, described in Figure 3, which allow for different levels of variability, ranging from level 1 up to level 6. The average number output by the dice is always 3.5 but the squared coefficient of variation ranges from 0.02 (level 1) up to 3.082 (level 6). Three of the dice types have six sides and three have 20 sides. Note, however, that the sides do not necessarily contain each value between one and 20, in order to correspond with the correct variation level and average. The player determines which die will be used and thus controls the variability in the production output of a station. One can choose the same die for all work stations or different dice at different work stations. This allows the simulation of lines with the same level of variability for all stations or lines with high/low variability at the start/middle/end of the line. Each roll of the die represents the potential production output of a work station during one period. This step can be repeated 50 up to 5000 times (periods).

Figure 4 depicts the input screen. The input part of this game is very easy. The player determines: (1) the type of die per work station; (2) the starting inventory at each work station; and (3) the maximum buffer content. After pressing the button SUBMIT and selecting the number of rolls (button ROLL) of the dice. A button click can either trigger the simulation of one new period, or it can also generate multiple periods (50 up to 5000) at once.

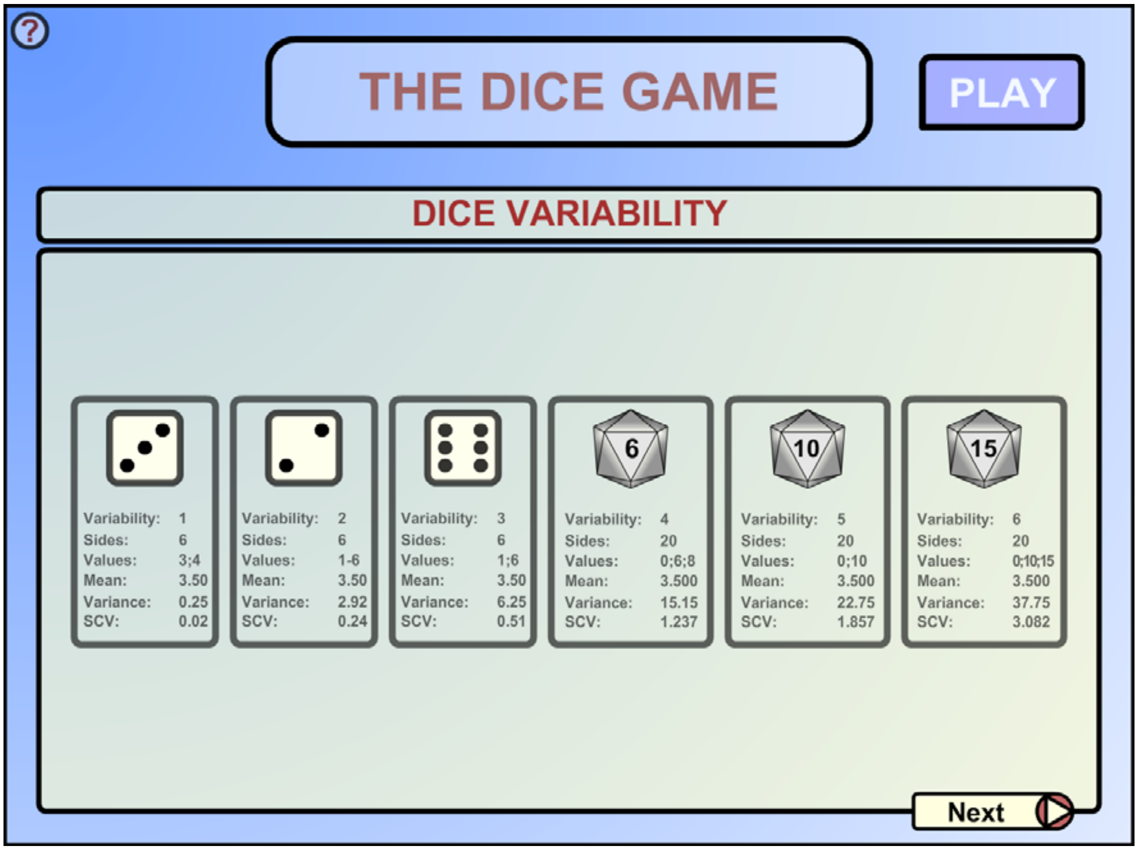

Figure 3 : The different types of dice

We advice the player to set the buffer size larger than or equal to the maximum face value of the die selected (in order to simulate the correct average value). It is also advisable to set initial inventory larger than or equal to the maximum buffer size (the steady-state condition will be reached faster in this way).

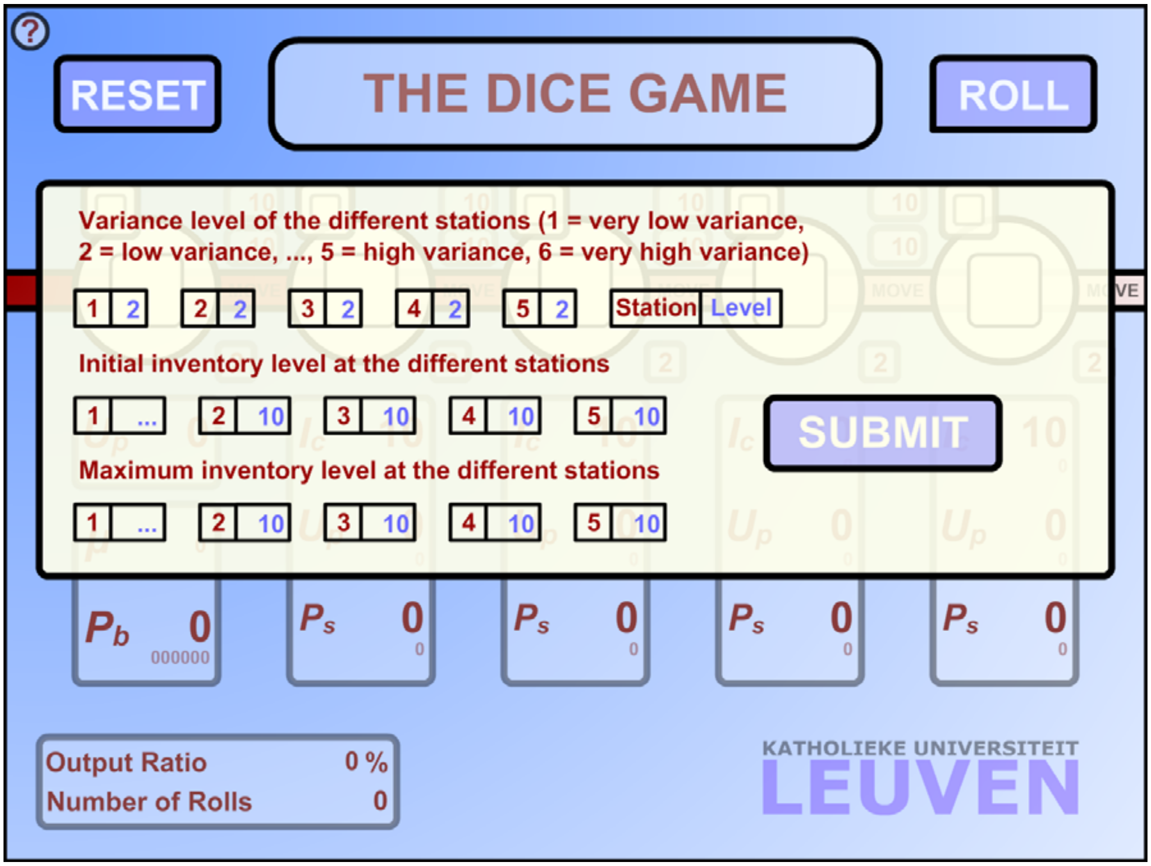

Figure 4 : The input screen

The die establishes the potential output of a work station. The actual output may differ because of starvation and blocking. We illustrate this important aspect of the game by means of an example described in Figure 5. Each station has an input buffer and an output buffer; the latter is actually the input buffer of the next station. We refer to the input buffer as the inventory of the workstation. Suppose that both buffers have an upper limit of eight units. The current inventory of the work station we will analyze equals four, the current inventory at the next station equals five. We roll the die and the outcome is six. Which actions are triggered?

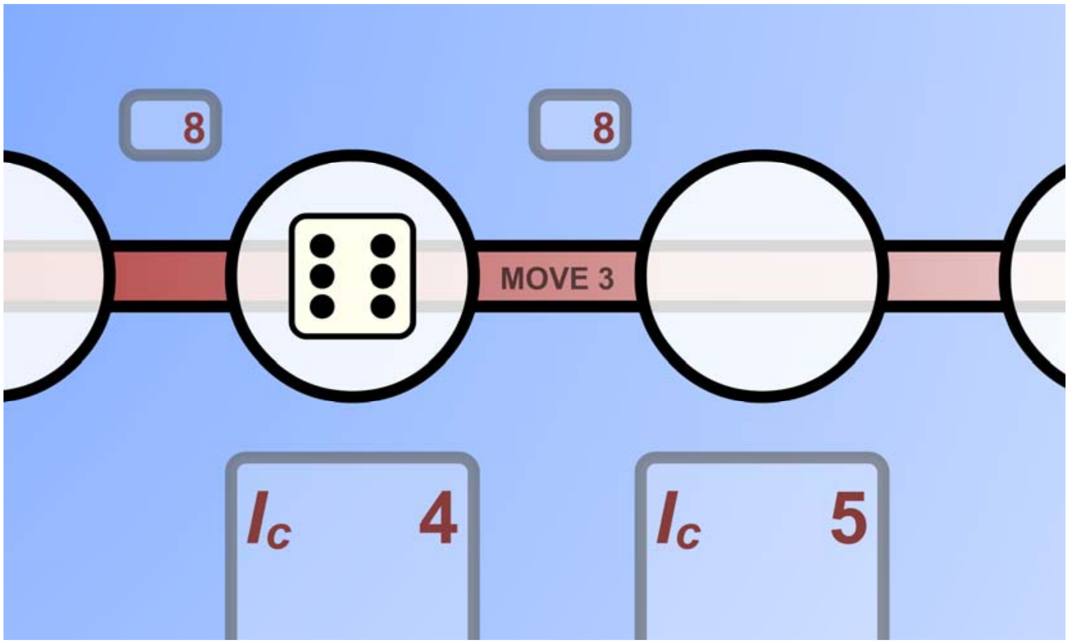

Figure 5 : Potential versus actual output

We obtained six as potential production quantity, but there are only four units available in the input buffer and consequently two units are starved. If we move the four units to the buffer of the next station then the inventory will become 9 (5+4), which is more than the maximum buffer size (which was set at 8). The number of units that will be moved (which is the actual output) is three – we lose two units because of starvation and one because of blocking.

These computations are performed for each work station at the beginning of each period. Once the actual output for each work station is known, the actual production output is transferred to the next station (the output of the last station serves the customer). All work stations operate simultaneously (i.e., concurrently). This is an important aspect of the dice game. Each run (period), the availability of work-in-process at the beginning of the period is checked, and it is not permitted to use material that was not available at the beginning of the period. In a sequential version of the game, the preceding operation's processed units are available to the subsequent operation *in the same period.*

We refer to Figure 6 for a discussion of the output of the game. We select type-2 dice, a starting inventory of ten units at each workstation and a buffer limit of ten units. The system is run for 1000 dice rolls.

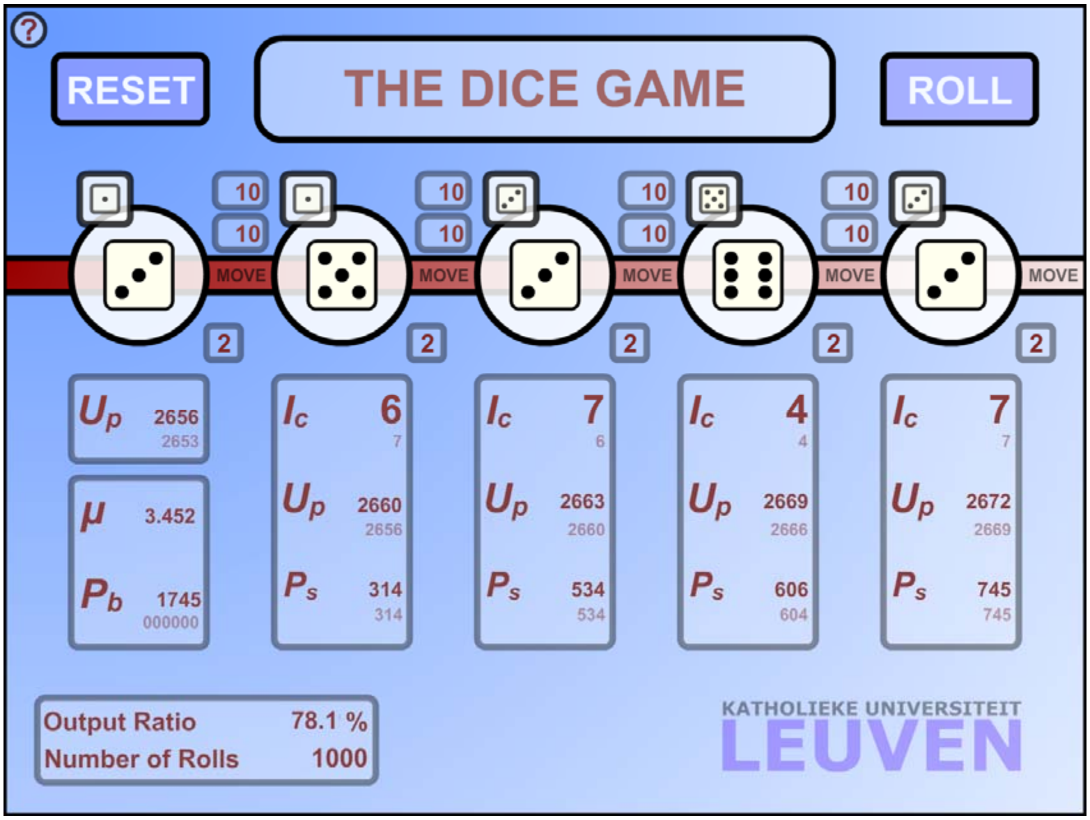

Figure 6 : Output of a simulation run

We use the following symbols in Figure 6. P_b stands for the total number of units blocked over all stations. P_s represents the number of units starved for a work station. U_p equals the actual output (units moved to the next station) and I_c indicates the current inventory (input buffer). Finally, μ stands for the average number of eyes of all dice rolls (grand average).

If we play the game 1000 times, we have a potential output of 3500 units (1000 × 3.5), but due to starvation and blocking the actual output will be far less than that number. Suppose, for example, that the actual output (number of units moved (completed) at the last working station) is 2672 and that the number of units starved at the last station equals 745 units. The total "demand" at the last station equals 2672+745= 3417 (the expected value is 3500 units). The output ratio consequently equals (2672/3417) = 0.781 or 78.1%. The difference between 3417 and 2672 is the output shortfall. The output ratio is our metric for measuring throughput.

As mentioned in the introduction, the Dice Game can be played as a computer-based simulation or as a hands-on manual game. A good layout for the manual game is given in Figure 1 (others may prefer a layout as given in Figure 2). The rules of the game are exactly the same as the ones explained above. While playing the game, information can be gathered and summarized in a table as given in Figure 7. The table in Figure 7 refers to work station 5. It is not necessary to collect data for the other work stations.

	Outcome of the die	Sent to the customer	# units starved
Period 1 **Period 2** **…**			
Sum		U_p	P_s
Output ratio	$\frac{U_p}{U_p + P_s}$		

Figure 7 : Data sheet for the manual game

The manual game will make the students familiar with the dice game and the step to the computerized version of the game will be made very easy. We do advise the instructor to play the game for at least 20 periods and to make clear to the students that the steady-state conditions will not be reached.

INSIGHTS FROM THE DICE GAME

The key learning element for students is the insight in the relationship between variability and throughput in an environment with dependent workstations and limited buffers. Limited buffers create starvation and blocking and this in turn impacts the throughput. Low-inventory environments are typically discussed and advocated in courses dealing with lean manufacturing. The dice game clearly illustrates the downside of so-called *lean* practices and it also shows that adding buffers can dramatically improve the throughput performance. The amount of "productive" buffer capacity depends on the degree of variability. This important insight can be experienced by students in a very simple way. Indeed, playing the dice game is easy and only requires a 15-minute introduction.

After playing the game manually, the simulation tool is used. We advise the instructor to start with a production line whose output potential is determined by a die of type 2. This die is a normal six-sided die (the number of eyes ranges from 1 to 6). We limit the buffer capacity to 10 and the beginning inventory is also 10. Subsequently, the students are asked to guess what the output ratio will be. 1000 rolls of the dice are simulated and an output ratio of approximately 78% will be obtained (which slightly varies from experiment to experiment). In other words, the throughput loss amounts to more than 20% and students will typically considerably underestimate this value. The experiment is then repeated with die type 1 (six sided but only with 3 or 4 as outcome), with the same buffer sizes. The output ratio will now be approximately 97.5%, which constitutes an impressive improvement. This small experiment usually stimulates students to go into greater detail.

The instructor can then introduce more experiments, preferably in small groups. Each group is assigned a die type and evaluates the output ratio for different buffer sizes. The instructor can collect the data and summarize the results, leading to a graph similar to Figure 8 (the y-axis refers to the output ratio and the x-axis refers to the buffer capacity).

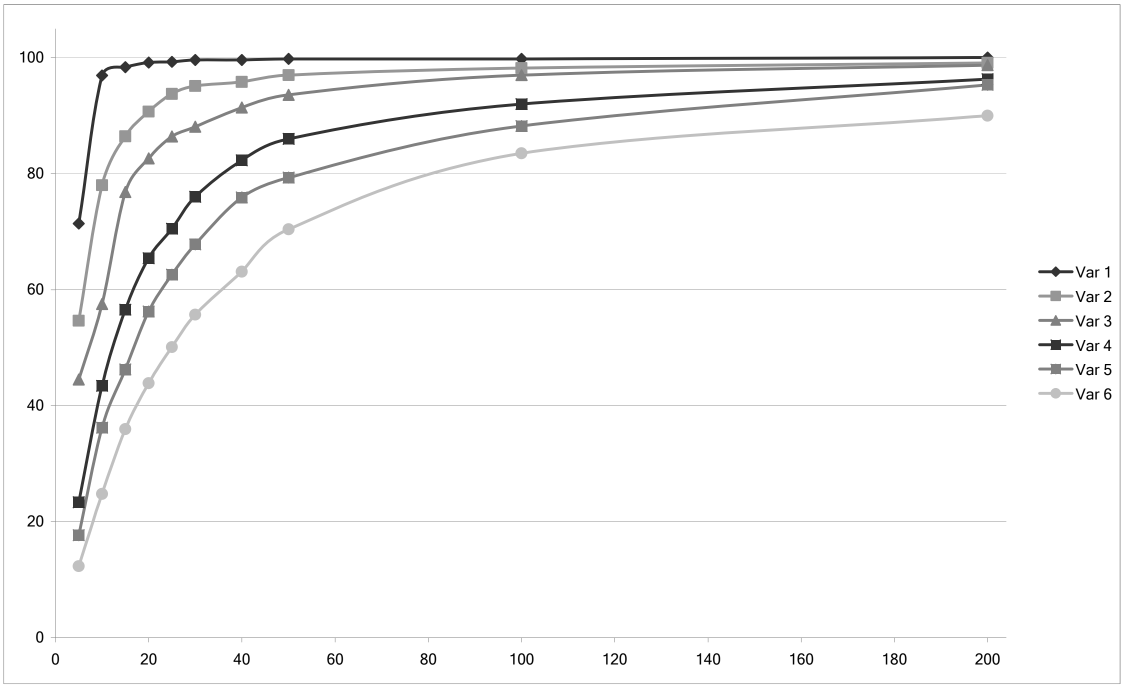

Figure 8 : The concave curve relating buffer size to output ratio

Figure 8 illustrates the relationship between buffer size (the same maximum buffer size for all work stations) and the output ratio for the different die types. For a buffer size of 20, for example, the output ratio ranges from 42% to 99%. Arriving at this point during the class sessions, it is advisable to introduce some theoretical insights explaining the simulation results. The paper by Conway et al. [7] is very helpful in that respect. Based on Figure 8 and [7] we can come up with the following insights. Note that these insights hold for balanced, equally buffered lines.

Insight 1 : The output ratio/buffer size curve is sharply concave. The more variability, the higher the throughput loss.

Insight 2 : To achieve a given output ratio, the buffer capacity should be proportional to the squared coefficient of variation of the processing rate.

Insight 3 : A little work-in-process helps a lot but diminishing returns arise. The rate of diminishing return depends on the level of variability.

Burman, Gershwin and Suyematsu [9] illustrate how helpful the above insights are in order to improve the design of a printer production line. In [10], the reader will find an excellent discussion on the corrupting influence of variability.

At this point in time the instructor can ask students what the major sources of variability are, and what can be done from an engineering point of view to eliminate these sources of variability. In this way, the game is positioned in a broader context.

Another interesting experiment is to find out what happens without an upper limit on the buffer size, which corresponds with the possibility only of starvation: there will be no blocking. By setting the maximum inventory level at 999 (see Figure 4) we can simulate a situation with only starvation. For the situation summarized in Figure 6, but with unlimited buffers, we obtain an output ratio of approximately 98%, compared to the 78% mentioned earlier with both blocking and starvation. This leads us to Insight 4.

Insight 4: In designing production lines, it is of crucial importance to determine the in-process inventory space. Limiting the buffer space may result in a dramatic output shortfall.

We refer to Gershwin [11] for an excellent theoretical treatment of the problem.

A final set of experiments that we recommend to perform in class is related to balanced lines (lines with the same expected output for all stations) with unequal variability. Up till now we have analyzed production lines with equal output potential both in terms of expected output and variability (by using the same die type for all work stations). In reality, it is very well possible that the variability of the output is different between the work stations. This can easily be simulated by means of the dice game: we simply select a different die type for each work station. Questions to be answered are the following: does this have an impact on throughput performance? Do we have to increase the buffer capacity to attain a target output ratio? Where is extra buffer space most effective?

One possible experiment proceeds as follows. We start with the standard situation described above, i.e., die type 2 and a buffer capacity of 10 units. The output ratio of this setting was said to typically be 78%. The die for work station three is then changed to type 5, while leaving the buffer capacities unaltered. The output ratio hereby drops to 55%. The following steps can subsequently be followed to re-attain the original target of 78%.
First increase the input buffer capacity of work station 3 to 20 units; no significant improvement will be observed. Next increase the output buffer capacity of work station 3 to 20 units as well. This will result in an output ratio of 74%. This outcome suggests that the strategy of increasing both the input as well as the output buffer is beneficial. Further improvement can be obtained by adding extra buffer space to stations 1 and 5. Take e.g. 15 units, the output ratio will increase to 78%. This suggests that it is not sufficient to protect the high-variability work station alone. We also have to protect the other work stations because the variability propagates over the whole line. A bell-shaped allocation of buffer capacity (centred around the high-variability work station) is in other words advisable (see [7]).

Insight 5 : Work stations with large variability have an impact on the output performance of the whole line. This means that the high-variability work station has to be protected by extra buffer capacity. Because of the propagation of variability the other work stations need extra protection as well. A bell-shaped allocation is advisable.

CONCLUSION

The dice game is a powerful learning exercise focusing on the impact of variability and dependency on throughput and work-in-process inventory of flow lines. In this paper we both offer a manual and a simulation-based tool to play the game. The insights obtained from the game have a great impact on the design of production lines. Engineering students should pay attention to the size of the in-process inventory space. The availability of buffers prevents the variability of each machine's production from blocking or starving the other work stations. Although these facts do not come as a surprise for most students, our experience indicates that students and managers usually underestimate the impact of variability.

Acknowledgments- The authors wish to thank the Fund for Scientific Research - Flanders, project G.0333.10N, for the financial support.

REFERENCES

1. E. Goldratt and J. Cox, The Goal: A process of Ongoing Improvement, 2nd Revised Edition, Great Barrington, Massachusetts, North River Press (1992).
2. L. Alarcón, and D. Ashley, Playing Games: Evaluating the Impact of Lean Production Strategies on Project Cost and Schedule, Proceedings of the 7th Annual Conference of the International Group for Lean Construction, p. 263-274 (1999).
3. O.H. Hilmola, Using Goldratt's dice-game to introduce system dynamics models and simulation analysis, Int. J. Information and Operations Management Education, Vol.4, 4, p. 363-376 (2006).
4. E.J. Umble and M Umble, The production dice game: an active learning classroom exercise and spreadsheet simulation, Operations Management Education Review, Vol. 1, p. 105-122 (2005).
5. A.C. Johnson and A.M. Drougas, Using Goldratt's game to introduce simulation in the introductory operations management course, INFORMS Transactions on Education, Vol. 3, 1, p. 20-33 (2002).
6. K.R. Baker, S.G. Powell and D.F. Pike, Buffered and unbuffered assembly systems with variable processing times, Journal of Manufacturing and Operations Management, Vol. 3, p. 200-223 (1990).
7. R. Conway, W. Maxwell, J.O. McClain and L.J. Thomas, The role of work-in-process inventory in serial production lines, Operations Research, Vol. 36, 3, p. 229-241 (1988).
8. Y. Dallery and S.B. Gershwin, Manufacturing flow line systems: a review of models and analytical results, Queuing Systems, Vol. 12, p. 3-94 (1992).

9. M. Burman, S B. Gershwin and C. Suyematsu, Hewlett-Packard uses operations research to improve the design of a printer production line, Interfaces, Vol. 28, 1, p. 24-36 (1998).
10. W.J. Hopp and M.L. Spearman, Factory Physics, McGraw-Hill International Editions, (2000).
11. S.B. Gerswin, An efficient decomposition method for the appropriate evaluation of tandem queues with finite storage space and blocking, Operations research, Vol. 35, 2, p. 291-305 (1987).

Marc Lambrecht is Professor of operations management, Research Center for Operations Management, at K.U.Leuven, Belgium. His research interests include stochastic modelling, inventory management and queuing models. He published over 35 journal articles in areas such as supply chain management, inventory optimization, queuing networks and the bullwhip problem. He recently published a paper (co-author Robert Boute) describing a simulation tool to explore the bullwhip effect.

Stefan Creemers holds a PhD in Applied economics on the topic of appointment-driven queueing systems (from K.U.Leuven, Belgium). He currently works as a postdoctoral researcher at the Research Center for Operations Management (K.U.Leuven, Belgium), is a visiting professor at HUBrussel (Belgium) and is owner of Sequentis, a consulting firm that specializes in project management solutions. His broad spectrum of research interests includes: supply chain management, project management and stochastic modelling.

Robert Boute is Assistant Professor at the Vlerick Leuven Gent Management School, Belgium. He holds an Msc degree in Commercial Engineering and a PhD in Applied economics from K.U.Leuven, Belgium. His research focuses on stochastic production/inventory systems and supply chain integration models. He recently published (co-author Marc Lambrecht) a paper describing a simulation tool to explore the bullwhip effect.

Roel Leus is Associate Professor of operations research at the Research Center for Operations Research and Business Statistics (ORSTAT) of K.U.Leuven, Belgium. Most of his research activities are within the domain of capacity planning and scheduling, in particular scheduling under uncertainty, project planning and scheduling and project portfolio management.