OPTIMIZATION CUSTOMIZED TOKEN-BASED PRODUCTION CONTROL SYSTEMS USING CROSS-ENTROPY[i]

Pedro L. González-R.
Industrial Management School of Engineering, University of Seville, Seville, Spain. pedroluis@esi.us.es
Framinan, Jose M.
Industrial Management School of Engineering, University of Seville, Seville, Spain. jose@esi.us.es
Dopfer, Andreas
Production and Industry Department. University of Duisburg-Essen. Duisburg. Germany. dopfer@uni-duisburg.de
Ruiz-Usano, Rafael
Industrial Management School of Engineering, University of Seville, Seville, Spain. usano@esi.us.es

One of the most recent approaches to production control mechanisms are the customized token-based production control systems, which have shown their effectiveness for different environments. Its optimization can be formulated as a combinatorial optimization problem. The cross-entropy method (CE) has confirmed to be useful for a great diversity of problems, including the NP-hard combinatorial optimization problems. However, the applicability of the CE algorithm for multi-criteria problems has not still been explored. In this paper we address the optimization of a customized token-based production control systems using the CE method for multi-criteria optimization. An algorithm for the optimization of these systems is proposed, while its performance is verified by a simulation model extracted from a component automobile factory in Spain.

1. INTRODUCTION

In the production control context, pull systems have been studied by researchers and successfully implemented in practice during last decades. Since the first Kanban approach used in Toyota factory (Monden, 1983) several pull systems (also termed as token-based systems by Gershwin, 2000) have been developed. The most general approach regarding token-based systems are the Customized Token-Based Systems (CTBS) (see e.g. Gaury *et al.*, 2000, or Gaury *et al.*, 2001, or González-R, 2006). CTBS try to control the maximum amount of jobs between each pair of stations by means of cards, similarly to the Kanban system which try to limit the maximum amount of work by each station. These systems seem to reach better results than other existing pull systems (see e.g. Gaury, 2000), while its main disadvantage is their correct customization, i.e. determining the control loops to be implemented and the number of cards for every control loop. This problem can be formulated as a combinatorial optimization problem. Besides, it is usual that more than one objective - such as throughput, work in process (WIP), or service level - must be considered. Therefore, the problem can be regarded as a multi-criteria problem. The Cross-Entropy (CE) method (Rubinstein, 1997) has shown to be useful for a great diversity of problems, including some NP-hard combinatorial optimization problems (Alon, *et al.*, 2005; De Boer, *et al.*, 2005; Rubinstein, 1999; Rubinstein and Kroese, 2004). However, to our best knowledge there are no references about the CE method regarding multi-criteria optimization. In this paper, we address the optimization of CTBS using the CE method for multi-criteria optimization.

The remainder of the paper is structured as follows. In section 2 the CTBS are described. Section 3 is devoted to briefly describe the CE method focused on the CTBS optimization. Section 4 presents an experiment showing the performance of the introduced method applied to a model based in a real manufacturing line. Conclusions and future research are outlined in section 5.

2. CUSTOMIZED TOKEN-BASED SYSTEMS

2.1 Token-based systems

In token-based systems (such as Kanban or Conwip), the flow of jobs through the shop is conducted by kanban cards - or tokens -. In a Single Kanban system there is a parameter k_i for every station i, meaning the maximum amount of work allowed for the corresponding station. Then, for one shop of n stations, the Single Kanban system requires the setting of n parameters (see figure 1).

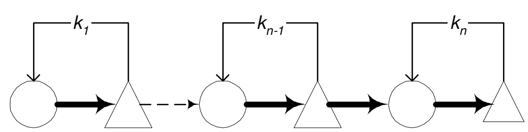

Figure 1 – Single Kanban production control system

The Conwip system uses one card count k for all stations in the line. Therefore, the maximum amount of work is limited by the number of cards and one only parameter must be set (see figure 2).

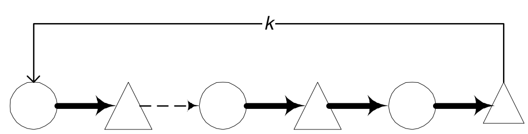

Figure 2 – Conwip production control system

Kanban systems use only local information about the state of every station. In the case of the Conwip system, it gets global information regarding the complete line, but not for sub-parts of the process. Both Kanban and Conwip are extreme cases of token-based systems. In the middle of these, there exist a great variety of token-based mechanisms, such as Base stock, Double Kanban, Extended Kanban, Generalized Kanban, Generic Kanban, or Hybrid Kanban-Conwip, among others. There are numerous references related to comparisons among token-based systems (see e.g. Framinan *et al.*, 2003), and results show that there is not one system that outperforms the others for all possible scenarios. The currently approach tries to implement one of the existing pull systems without any assurance about its fitness for that environment. Therefore, it would be of interest the design of an *ad-hoc* token-based system for a given shop and manufacturing conditions. This is the approach carried out by the CTBS, which is explained in the next subsection.

2.2 Customized token-based systems description

If we consider all possible card loops - control loops - between all stations in the system, we have the most general token-based system, such as the one shown in figure 3. This type of system is known as Customized Token Based System (CTBS). Every existing token-based system is a particular case of the CTBS. For example, if we only consider the card loop between the first and the last station (and set a high value for the cards in the rest of the loops), we obtain a Conwip system.

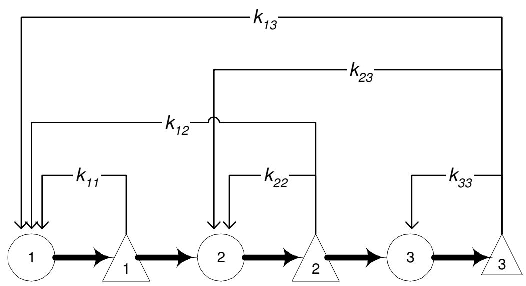

Figure 3 – 3-Station CTBS before optimization.

Hence, one CTBS will initially consider every possible loop between stations. An ‘optimization’ process has to be then carried out in order to simultaneously remove those loops not useful for the specific scenario and to set the proper number of cards for the remaining loops. It has been empirically established that only less than the half of the initial loops is usually necessary to be implemented (Gaury, 2000).

2.3 Search space

The advantages of the CTBS against other ‘traditional’ pull system have been studied in different works (see e.g. Gaury 2000, or González-R. 2006). Nevertheless, the main problem coupled to CTBS lies in their optimization. This can be formulated as finding a correct integer value for every possible loop according to a set of objectives. If we initially consider all possible card loops between two stations, the total number of loops, m, in one line formed by n stations is:

$$m = n\ (n+1)/2 \tag{1}$$

The total search space is given for all combinations of the values assigned to the loops. For a fixed number of cards, K_{max}, for all loops, the expression holds:

$$(K_{max})^m \tag{2}$$

The usual way to compute a solution in these systems is by means of discrete event simulation. Therefore, in one hand, the completely enumeration of all possible solutions makes the optimization problem intractable. On the other hand, alternative methods, as mathematical modeling or queuing theory can be only useful for very simplistic cases and not appropriate in most real environments. Hence, heuristic algorithm can be a realistic approach for addressing the optimization problem.

2.4 Structural properties

In last sub-section, it was pointed out that one of the most important problems for the CTBS optimization was the search space size. The 'structural properties' can be used in order to efficiently reduce the search space. Although this is not the aim of this paper it is important to introduce its concept, in order to clarify later sections. Structural properties for CTBS were proposed by Gaury (2000) and later generalized by González-R. (2006). Similar works have been done for Kanban systems (see e.g. Tayur 1993 and Ramesh 1997). These properties are termed structural properties, because there is no dependence with particular characteristics of the line, such as time processes, existence of machine breakdowns, set-up times or demand behavior, but for the number of stations. Then, it is possible to detect if one particular solution is a structural efficient solution or not. The non-structural efficient solutions are not useful to be explored because the Pareto set of efficient solutions is included in the set of structural efficient solutions (González-R., 2006). Structural properties try to detect for a specific solution which loops do not affect the performance of the system, because of they do not limit the work flow through the line. Then, structural properties can be used in two different ways: first to simplify a certain solution and second to generate structural efficient solutions. These aspects will be taken into account in the optimization procedure described in next section.

3. THE CROSS-ENTROPY METHOD

3.1 Description

Many problems in Operations Research try to solve very complicated optimization problems. Commonly, the complexity of these problems is known in advance and the objective function can be evaluated using a closed-form expression. However, there are other types of problems where there is not a known expression for the objective function. In these cases, the objective function must be estimated. Discrete event simulation is a common way to estimate the value of an unknown objective function expression. The CE method (see e.g. Rubinstein and Kroese, 2004) provides a simple and efficient method to solve those types of problems.

The CE method is an iterative process formed by two steps:

1. Generate a set of random data (e.g. variables, or vectors) according to a specific mechanism.
2. Update the parameters of the mechanism in order to produce a better set in the next iteration.

It is possible to show that this method reaches the global optimum for certain cases. Another important aspect is that the CE method provides a unified scheme for simulation and optimization problems. These characteristics make the CE method a reasonable technique to be used in the CTBS optimization.

3.2 CE applied to optimize CTBS

CTBS can be characterized by a vector of cards, $\mathbf{k} = \{k_1, k_2, ..., k_i, ..., k_m\}$, formed by m components given by the maximum number of loops - see equation (1) -. The possible values for every vector component are integer numbers, which represent the number of cards in that loop. Alternatively, it is possible to delete some loops. In that case the infinite value is assigned to it. Hence, the domain for the vector components is $1 \le k_i \le K_{\max} \cup \infty$.

Every vector of cards will produce different values of throughput/service level and WIP in the system. Then, the problem can be interpreted as a combinatorial optimization problem. The notation used in order to explain the main algorithm steps are the following:

$S(\mathbf{k})$, system performance, objective function

$\hat{\gamma}$, minimal expected throughput for the elite solution

N, population size

ρ, population fraction for elite solutions, usually bounded such, $0.01 \le \rho \le 0.03$ in case that the number of components in the vector is greater than 100, in other case $\rho \approx \ln(n)/n$.

$\mathbf{k}^{(y)}$, certain elite solution vector

$\mathbf{Q}$, is a probability matrix for random set generation

α, smoothing parameter

The main CE algorithm for the optimization of the CTBS can be implemented by the following steps:

0. Generate a random set of N cards vectors, $\{\mathbf{k}^{(1)}, \mathbf{k}^{(2)}, ..., \mathbf{k}^{(i)}, ..., \mathbf{k}^{(N)}\}$
1. Select a group of elite solutions according a predefined criterion (e.g. throughput maximization, $\max S(\mathbf{k})$). Usually is done selecting a fraction ρ of the population, i.e. ρN.
2. For each elite solution: count for every component in the vector the percentage of the number of times that a certain value is repeated. Later will be used in order to obtain an idea of those better values for every component in one vector. It is usually to make use of a $\mathbf{Q}$ matrix, where every component q_{ij} means the percentage of times that a particular

component k_i in a certain elite solution $\mathbf{k}^{(y)}$ has the value j. In our implementation the infinite value is determined by $j = 0$. $\mathbf{Q}$ matrix can be interpreted as a probability matrix for better solutions.

3. Using $\mathbf{Q}$ matrix, generate a new population set and go to step 1.

The stop criterion is usually defined by the convergence of the expected response in the elite group solutions, $\hat{\gamma}$ parameter. It is to note that, in order to improve performance of the system, better solutions can be obtained by using the structural properties every time a solution is generated. Consequently, the performance of the algorithm is increased by using 'structurally efficient solutions' on every population generation. It is a good practice to smooth the $\mathbf{Q}$ matrix, with the aim to avoid that probabilities for some components q_{ij} tend to zero or one at an early iteration. Then, it is common the use of the smoothed value of $\mathbf{Q}$, such as is expressed in (3) for the current iteration t.

$$\mathbf{Q}_t = \alpha \mathbf{Q}_{t-1} + (1-\alpha)\mathbf{Q}_t \tag{3}$$

Common values for the smoothing parameter, α, are between 0.7 and 1. Usually the initial population is obtained by assigning the same probability for every component. In our case it could be obtained by $1/(K_{\max} + 1)$.

3.3 CE multi-criteria approach

Gaury (2000) shows three useful objective functions for the case of reaching a target throughput/service level with the smaller WIP, by a lexicographical approach. However, the set of Pareto efficient solutions, Ψ, is sometimes required. In this sub-section is described an efficient method in order to obtain the Pareto frontier, based in the CE method described in the last section. The procedure consists on iteratively updating the Pareto set, marking explored solutions as visited. The method is handled by the following steps:

0. Generate a random solution $\mathbf{k}^{(0)}$ and update the set of Pareto efficient solution, Ψ.
1. Choose the first non-marked as visited solution $\mathbf{k}^{(i)}$ from the Pareto efficient solution set, Ψ, (under some sorting criterion), and set the output as the target throughput/service level, $\theta = S(\mathbf{k}^{(i)})$. If Ψ is empty, generate a random solution. Mark $\mathbf{k}^{(i)}$ as visited solution.
2. Use the CE algorithm described in 3.2 in order to reach the target throughput/service level, θ updating the set of efficient solutions, Ψ.
3. Go to step 1 until all the Pareto set, Ψ, is marked as visited or other stop criterion is reached.

The sorting criterion we employed was from minimum throughput to maximum throughput. Additional stop criterion could be established, such us maximum number of iterations. This algorithm makes an iterative use of the CE method, by means of an intensive search of better solutions on every new solution in the Pareto set. It is important to notice that while the algorithm is trying to converge to a certain target value, the probability matrix also explore the neighborhood on every solution. Therefore, this searching procedure is profitable on identifying new Pareto frontier points.

4. COMPUTATIONAL EXPERIENCE

In order to compute the algorithm effectiveness described in sub-section 3.3, we conduct an experiment based on real data of an automobile component factory from Spain. Although detailed data (provided by the production control department of the factory) are not included in this paper for confidentiality reasons, we mention the main characteristics of the line. The experiment was focused on a ten-station line in tandem. Processing times can be considered deterministic, being affected by different variability causes. The model considers 13 important station breakdowns, with mean times between failures (MTBF) exponentially distributed, while mean times to repair (MTTR) follow an Erlang distribution. Additionally, 23 scheduled maintenance labors have been considered. Their frequency and interval have been set as deterministic. We also consider 15 micro-pauses with MTBF exponentially distributed while MTTR occur in a deterministic interval. A simulation horizon of 40 days and 10 replicates were used for every experiment, in order to get reliable measures. In figure 5 the Pareto efficient solution for the CTBS using the proposed CE methodology for multi-criteria problems is shown. The figure also illustrates those results obtained for the same line if managed by a Conwip system.

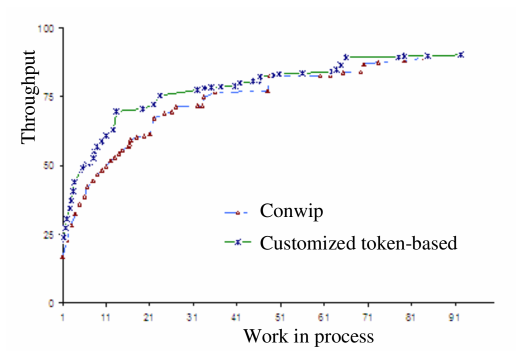

Figure 5 - Pareto frontier for the CTBS and Conwip solution.

Currently, the production line in the factory is controlled by means of a Kanban system, reporting a throughput level of 90% and a WIP of 350 jobs. In contrast, the customized system can reach a 90.17% throughput with only 92.33 WIP. The customized system for that throughput is shown in figure 6. Circles means machines, while triangles stand for buffers.

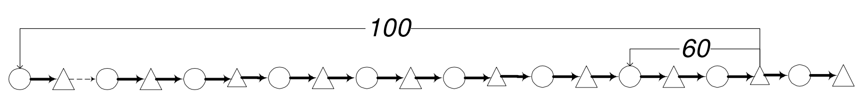

Figure 6 - Customized system for a 90.17% of throughput.

It is to note that the proposed CTBS is easier to control and put into practice, because it only depends on two parameters, in contrast to the ten parameters used in the currently implemented Kanban system.

5. CONCLUSIONS

In this work we have introduced the CTBS and presented it as an efficient alternative for existing pull mechanisms. In section 3 we have briefly described the CE method adapted to the optimization of the CTBS, under a multi-criteria approach.

In order to test the performance of the proposed optimization algorithm, we test it in a model based on real data of a component automobile factory. Results show that the resulting CTBS reaches better results than a Conwip system for every combination of throughput and WIP. Additionally, it outperforms the production control system currently under operation in the factory.

6. REFERENCES

1. Alon, G., Kroese, D.P., Raviv, T., Rubinstein, R.Y. Application of the Cross-Entropy Method to the Buffer Allocation Problem in a Simulation-Based Environment. Annals of Operations Research 2005; 134(1): 137-151.
2. De Boer, P-T., Kroese, D.P., Mannor, S., Rubinstein, R.Y. A Tutorial on the Cross-Entropy Method. Annals of Operations Research 2005; 134(1): 19-67.
3. Framinan, J.M., González-R, P.L. and Ruiz-Usano, R. The Conwip production control system: review and research issues. Production Planning and Control 2003; 14(3): 255-265.
4. Gaury, E.G.A. Designing pull production control systems: customization and robustness (PhD Thesis, Ed. Center). Tilburg University, The Netherlands, 2000.
5. Gaury, E.G.A., Pierreval, H., Kleijnen, J.P.C. An evolutionary approach to select a pull system among Kanban, Conwip and Hybrid. Journal of Intelligent Manufacturing 2000; 11: 157-167.
6. Gaury, E.G.A., Pierreval, H., Kleijnen, J.P.C. A methodology to customize pull control systems. Journal of Operational Research Society 2001; 52(7): 789-799.
7. Gershwin, S.B., Design and operation of manufacturing systems_ the control-point policy. IIE Transactions 2000; 32:891-906.
8. González-R., P.L. Design and operation of Customized token-based production control systems. (PhD Thesis). University of Seville, Seville, Spain, 2006.
9. Monden, Y. Toyota production system. Industrial Engineering and Management Press, Atlanta, 1983.
10. Ramesh, R., Prasad, S.Y., Thirumurthy, M.V. Flow control in a Kanban based multicell manufacturing: I. A structural analysis of general blocking design space, International Journal of Production Research 1997; 35(8): 2327-2343.
11. Rubinstein, R. Y. Optimization of computer simulation models with rare events. European Journal of Operations Research 1997; 99: 89-112.
12. Rubinstein, R. Y. The cross-entropy method for combinatorial and continuous optimization. Methodology and Computing in Applied Probability 1999; 2: 127-190.
13. Rubinstein, R.Y. and Kroese, D. P. The Cross-Entropy Method: A Unified Approach to Monte Carlo Simulation, Randomized Optimization and Machine Learning. Information Science & Statistics, Springer Verlag, NY, 2004.
14. Tayur, S.R. Structural properties and heuristic for Kanban-controlled serial lines. Management Science, 1993; 39(11): 1347-1368.

---

[i] This research has been supported by the Spanish National Research Plan under DPI2004-02902 project.