AN INTEGRATED BINARY-TABU SEARCH APPROACH FOR THE BUFFER ALLOCATION PROBLEM: AN INDUSTRIAL CASE STUDY

Leyla Demir^(a)^, **Simge Yelkenci Köse**^(b)^, **Semra Tunalı**^(c)^, **Deniz Türsel Eliiyi**^(d)^

(a) Pamukkale University, Department of Industrial Engineering, 20070, Denizli, Turkey
(b) Ministry of Science, Industry and Technology Directorate General for Industry, Çankaya-Ankara, Turkey
(c)Izmir University of Economics, Department of Business Administration, Sakarya Cad. No:156, Balcova-Izmir, Turkey
(d)Izmir University of Economics, Department of Industrial Systems Engineering, Sakarya Cad. No:156, Balçova-Izmir, Turkey

(a)ldemir@pau.edu.tr, (b)simge.yelkenci@deu.edu.tr, (c)semra.tunali@ieu.edu.tr, (d)deniz.eliiyi@ieu.edu.tr

ABSTRACT

In this study, an integrated binary-tabu search algorithm is proposed to solve the buffer allocation problem for a real manufacturing system producing heating exchanger in Turkey. The aim is to minimize the total buffer size in the system while improving the system performance. To achieve this objective first the production system is modeled by using simulation. After that the proposed algorithm is employed to find the optimal buffer levels for minimizing the total buffer size in the system. The experimental study shows that proposed algorithm improves the current average daily throughput rate about 31.68%.

Keywords: buffer allocation problem, simulation, tabu search, combinatorial optimization

1. INTRODUCTION

The buffer allocation problem is an NP-hard combinatorial optimization problem which involves the distribution of buffer space among the intermediate buffers of a production line. This problem arises in a wide range of manufacturing systems. The primary effect of storage buffers is to allow machines to operate nearly independently of each other. Storage buffers help reducing idle time due to starving (no input available) and blocking (no space to dispose of output). Less idle time increases average production rate of the line. However, inclusion of buffers requires additional capital investment and floor space. Buffering also increases in-process inventory. If the buffers are too large, the capital cost incurred may outweigh the benefit of increased productivity. If the buffers are too small, the machines will be underutilized or demand will not be met. Because of the importance of finding optimal buffer configuration, the buffer allocation problem is still an important optimization problem faced by manufacturing system designers.

The buffer allocation problem is solved by employing evaluative and generative methods in an iterative manner. Evaluative methods are used to calculate the performance measures of the system, such as throughput rate and work in process levels. The decomposition method (Gershwin and Schor 2000; Helber 2001; Shi and Men 2003; Tempelmeier 2003; Shi and Gershwin 2009; Demir, Tunali and Løkketangen 2011; Demir, Tunali and Eliiyi 2012), aggregation method (Diamantidis and Papadopoulos 2004; Dolgui, Eremeev, Kolokolov and Sigaev 2002; Dolgui, Eremeev and Sigaev 2007; Qudeiri, Yamamoto, Ramli and Jamali 2008), generalized expansion method (Cruz, Duarte and Van Woensel 2008; Aksoy and Gupta 2010) and simulation (Lutz, Davis, and Sun 1998; Jeong and Kim 2000; Gurkan 2000, Sabuncuoglu, Erel and Kok 2002; Sabuncuoglu, Erel and Gocgun 2006; Bulgak 2006; Altiparmak, Dengiz and Bulgak 2007; Battini, Persona and Regattieri 2009; Can and Heavey 2011; Can and Heavey 2012; Amiri and Mohtashami 2011) are the most widely used methods among them. The first three methods are approximate analytical methods which are applicable under certain assumptions. To overcome these restrictive assumptions so that we could model our real system realistically simulation is used in this study. Generative methods are used for optimizing decision variables, such as buffer levels and service rate. There are various techniques used as generative methods, such as dynamic programming (Yamashita and Altiok 1998; Diamantidis and Papadopoulos 2004), gradient search method (Seong, Chang and Hong 2000; Gershwin and Schor 2000; Helber 2001), and meta-heuristics (Lutz, Davis, and Sun 1998; Spinellis and Papadopoulus 2000a; Spinellis and Papadopoulus 2000b; Spinellis, Papadopoulos and MacGregor Smith 2000; Dolgui, Eremeev, Kolokolov and Sigaev 2002; Dolgui, Eremeev and Sigaev 2007; Shi and Men 2003; Nourelfath, Nahas and Ait-Kadi 2005; Nahas, Ait-Kadi and Nourelfath 2009; Demir, Tunali and Løkketangen 2011; Demir, Tunali and Eliiyi 2012; Amiri and Mohtashami 2011; Can and Heavey 2011; Can and Heavey 2012).

In this study, we propose a new generative method to solve the buffer allocation problem which integrates binary search and tabu search methods. In our solution methodology, while the binary search is used as an outer control loop algorithm to minimize the total buffer size in the system tabu search is used as an inner loop algorithm to optimize the buffer levels for a given total buffer size. The proposed solution approach is employed for optimizing the buffer levels in a heating exchanger production system. The ultimate aim is to improve the average daily throughput rate in this heating exchanger system by integrating the proposed solution methodology with simulation model of the system.

The rest of this paper is organized as follows. In the next section the simulation model of the system is described in detail. The details of the proposed binary-tabu search algorithm are given in Section 3. Section 4 presents the results of computational study. Finally, concluding remarks and some future research directions are given in Section 5.

2. SIMULATION MODEL

The proposed approach is implemented to solve the buffer allocation problem in a local company operating in Izmir, Turkey. The company is produce-to-order type and produces seven types of heating exchangers. As it is seen in Figure 1, the production line is composed of pressing, forming, welding, testing, packaging and assembly stations with parallel machines. First, based on customer specifications, the steel rolls are smoothed and cut. Next, using different types of moulds, the steel rolls are processed in batches at the press station. After the press operation, each part is directed to subassembly stations where four machines operate in parallel. Then as shown in Figure 1, the parts visit the oven, welding, test and packaging stations.

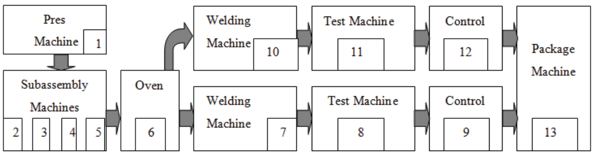

Figure 1: Production flow

The detailed simulation model of this line is developed in Arena 10.0 under the following assumptions:

- The plant operates three shifts per day including 50 and 15 minutes breaks for meal and cleaning, respectively in each shift. Each shift takes 8 hours, so the simulation model is run for 1245 minutes, corresponding to one working day.
- Raw materials are always available and there is sufficient space to store the finished products.
- The capacity to store work-in-process between the machines is limited.
- Each machine is operated by one operator, except for the press (1) and welding machine (7) (see Figure 1), which require 2 operators.
- Transporters are used for material handling between the machines in the system.
- Six different types of parts (LM1 to LM6) are pressed and then combinations of these are assembled to form seven types of heating exchangers. For example, type 1 heating exchanger is formed by an assembly of LM4 and LM5.
- Removing one type of mould from a press machine and replacing with another type requires a setup operation. To fit an appropriate distribution, the setup time records kept in the company are analyzed using Input Analyzer of ARENA 10.0 and the resulting estimates of input distributions are given in Table 1.
- Processing times are stochastic, except for the processing times on the automated press machine. Each machine is subject to random breakdown. The time-to-failure and the repair time for each machine are exponentially distributed.

Table 1: The setup times at the press machine

Parts	Setup Times (in minutes)					
	LM1	LM2	LM3	LM4	LM5	LM6
LM1	---------	U(41,47)	U (27,33)	U (32,38)	U (42,48)	U (37,43)
LM2	U(27,33)	---------	U (28,34)	U (27,33)	U (27,33)	U (42,48)
LM3	U (27,33)	U (39,45)	---------	U (32,38)	U (42,48)	U (27,33)
LM4	U (32,38)	U (35,41)	U (32,38)	---------	U (35,41)	U (35,41)
LM5	U (27,33)	U (32,38)	U (34,40)	U (27,33)	---------	U 35,41)
LM6	U (35,41)	U (27,33)	U (29,35)	U (27,33)	U (27,33)	---------

The model verification has been done by developing the model in a modular manner, using the interactive debuggers, and manually checking the results. The simulation model is validated by comparing the output of the simulation (i.e., daily throughput) with the output of the real system at 95% confidence level and the simulation model has been found to be truly representing the real system. Since the company operates on the basis of produce-to-order and starts production without any work-in-process, the warm-up period has not been considered during the experiments. Besides validating the simulation model with respect to daily throughput, we carried out further experimental studies to estimate average machine utilizations and average number in queue for each machine. As a result of these studies, it is noted that the machines 4, 5, 6 and 7 have the highest utilizations and highest number of parts waiting to be processed.

Solving this real-world buffer allocation problem involves twelve decision variables denoting the buffer sizes to allocate to each station. The objective is to determine the best buffer configuration so that the capacity of the production line can be improved with minimum total buffer size. The current average production rate of this line is 95 heating exchangers per day and the company officials desire at least 30% improvement in the production rate. Therefore, the rate of 0.10, i.e. 125 heating exchangers per day, is defined as the desired throughput rate (f^*) of the real production system.

3. THE PROPOSED INTEGRATED BINARY-TABU SEARCH ALGORITHM

The proposed integrated binary-tabu search (B-TS) approach has two control loops, i.e., the inner loop and outer loop. While the inner loop control includes a tabu search (TS) algorithm, a binary search (BS) algorithm is employed as an outer loop. These nested loops aim at minimizing the total buffer size to achieve the desired throughput level. Figure 2 summarizes the execution mechanism of the proposed integrated B-TS approach. The outer loop, i.e. the BS algorithm, is started with a pre-specified N value, and then the maximum throughput rate that can be obtained with this N value is calculated using the TS algorithm. This throughput rate is then compared to the desired throughput rate, and new N values are suggested by the BS algorithm in an iterative manner. The procedure continues until the termination criterion of the BS algorithm is satisfied, i.e., until the desired throughput rate is achieved with the minimum total buffer size. The following sections present the details.

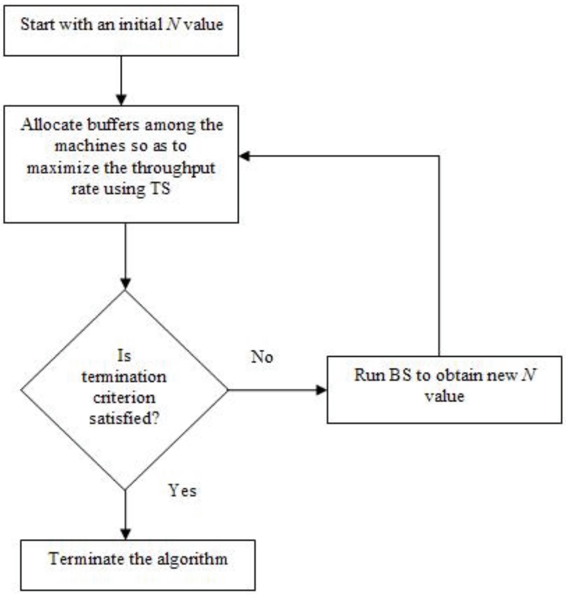

Figure 2: The framework of the integrated B-TS algorithm

3.1. Binary Search

Table 2 represents the steps of our systematic binary search. The algorithm starts by setting the upper and lower limits of total buffer size to M (a large value) and zero, respectively. The search continues between H and L until the desired throughput rate is obtained.

In this algorithm, $f(B^N)$ represents the throughput rate obtained with total buffer size N, f^* represents desired throughput level, and B^N represents buffer configuration obtained by total buffer size N.

Table 2: Pseudo code of the BS algorithm

```
0. (Initialization). Set $H = M$, $L = 0$.
1. Set $N = (H + L)/2$, $N^{\min} = N$.
2. Run TS algorithm for N.
3. If $f(B^N) < f^*$, set $L = N$ and $N = (H + L)/2$,
   If $f(B^N) \geq f^*$, set $H = N$, $N^{\min} = N$ and $N = (H + L)/2$.
4. Until $H = L$ go to Step 2, otherwise stop.
```

3.2. Tabu Search

Tabu Search is a meta-heuristic method for solving combinatorial optimization problems. TS explicitly uses the history of the search, both to escape from local optima and to implement an explorative search. For more details about TS the reader can refer to Glover and Laguna (1997). The following sections explain the features of the TS algorithm adopted in this study. It should be noted that this algorithms is similar to the one in Demir, Tunali and Løkketangen (2011), except that in this study TS is integrated with BS algorithm.

3.2.1. Move representation and tabu moves

In the proposed TS algorithm, the moves are represented by $[i, j]$, where i shows the location that a given amount of buffer is added and j shows the location that the same amount of buffer is subtracted. The increment (decrement) value is set to 1. Once a move is realized, the reverse of this move is recorded as tabu. Namely, if the move $[i, j]$ produces the best solution for the current step, then the reverse move $[j, i]$ is considered as a tabu for a certain number of iterations.

3.2.2. Search space and neighborhood structure

The search space of TS is simply the space of all feasible solutions that can be visited during the search. To define the neighborhood of the current solution, there are several choices depending on the specific problem at hand. In the buffer allocation problem context, one choice could be to consider the full neighborhood of the current buffer configuration while another could be to consider only a subset of the neighborhood (Demir, Tunali and Løkketangen 2011). In this study, all feasible solutions are considered as the search space, and all neighbors of the current solution are created and evaluated.

3.2.3. Tabu tenure

The length of the tabu list, called tabu tenure (TT), is the number of iterations that tabu moves stay in the tabu list. As indicated by Glover, Taillard and Wera (1993) the size of the tabu list providing good results often grow with the size of the problem. However, no single rule has been found to yield an effective tenure for all classes of problems. If the size of the tabu tenure is too small, preventing the cycling might not be achieved; conversely a long length may create too many restrictions. As indicated by Reeves (1996), a value of 7 for tabu tenure (*TT*) has often found to be sufficient to prevent cycling. Considering the problem size and after pilot experiments, *TT* is set to 7 throughout the algorithm.

3.2.4. Intensification

The key idea behind the concept of intensification is to implement some strategies so that the areas of the search space that seem promising can be explored more thoroughly. In general, intensification is based on a recency memory, in which one records the number of consecutive iterations that various solution components have been presented in the current solution without interruption (Demir, Tunali and Eliiyi 2012). Intensification is used in many TS implementations, but it is not always necessary. In this study, since the increment (decrement) values of the moves is set to 1 there is no need to spend time exploring in depth the portions of the search space that have already been visited, so an intensification strategy is not implemented.

3.2.5. Diversification

Diversification guides the search to unexplored regions to avoid local optimality, and it is usually based on a frequency memory where the total number of iterations of the performed moves or the visited solutions is recorded. There are two major diversification techniques known as restart diversification and continuous diversification. While the first is performed by several random restarts, the latter is integrated into the regular search process (Demir, Tunali and Eliiyi 2012). To improve the search process, the random restart diversification method is employed in the proposed TS algorithm. For the restart diversification, if the value of objective function does not change for a certain number of iterations, a random restart is applied. This value is set to 50 throughout the experiments.

3.2.6. Aspiration and termination criterion

In our TS algorithm, the tabu move is allowed if it results in a solution with an objective value better than the incumbent solution. The algorithm is terminated after a fixed number of iterations. This value is set to 1000 in our experiments.

The next section explains the implementation of the proposed solution approach in detail.

4. AN INDUSTRIAL CASE STUDY

The proposed solution methodology is employed to solve the buffer allocation problem in a heating exchanger production system explained in Section 2. The problem is mathematically formulated as follows:

Find $B = (B_1, B_2, ..., B_{K-1})$ so as to

$$\min\ N = \sum_{i=1}^{K-1} B_i \tag{4}$$

subject to

$$f(B) \geq f^* \tag{5}$$

$$B_i \text{ nonnegative integers } (i = 1, 2, ..., K-1) \tag{6}$$

$$0 \leq B_i \leq u_i \quad i = 1, 2, ..., K-1 \tag{7}$$

where N is a fixed nonnegative integer denoting the total buffer space available in the system that has to be allocated among the K-1 buffer locations so as to improve the performance of the K-machine production line. In this formulation, B represents a buffer size vector, B_i is the buffer size for each location, $f(B)$ represents the throughput rate of the production line as a function of the buffer size vector, f^* is the desired throughput rate, and u_i represents the upper bounds for each buffer capacity.

Figure 3 shows the framework of the proposed solution approach explained in previous sections. In this configuration simulation model of the production system is used to obtain the average daily throughput, i.e. production rates. This throughput rate is then compared to the desired throughput rate, and a new buffer configuration is suggested by the proposed B-TS algorithm in an iterative manner. The procedure continues until the termination criterion of the algorithm is satisfied, i.e., until the desired throughput rate is achieved with minimum total buffer size.

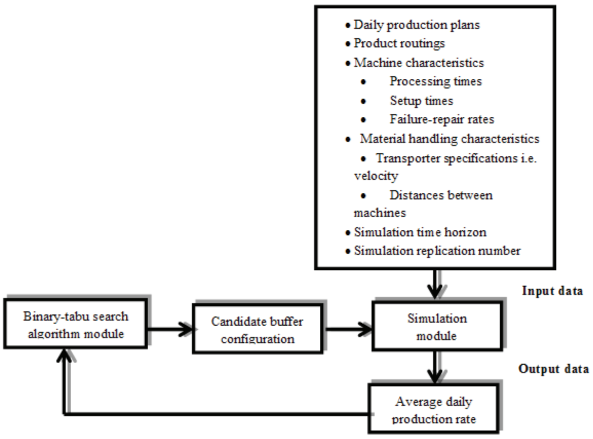

Figure 3: The framework of the proposed solution procedure for buffer allocation problem

The integrated binary-tabu search algorithm is implemented in C language. The experiments are carried out on a PC with 2 Ghz Pentium (R) 4 CPU processor, 2 GB RAM.

B-TS reaches the minimum total buffer size after 7 iterations (totally 7000 iterations as each iteration of BS algorithm involves 1000 iterations of TS algorithm). The minimum total buffer size is achieved with the buffer configuration $B = \{1,2,1,1,2,3,1,3,1,3,1,1\}$. The corresponding average daily throughput rate and the total buffer size for the proposed solution are 0.10048/minutes, 20 units, respectively. This means B-TS improves the current average daily throughput rate about 31.68 % for the system. So, it can be concluded that the proposed B-TS algorithm is very efficient to solve buffer allocation problem in a real manufacturing environment.

5. CONCLUSION

In this study, an integrated binary-tabu search algorithm is proposed to solve a real-world buffer allocation problem. This algorithm is integrated with a simulation model that captures the stochastic and dynamic nature of the production line. The simulation model is used to calculate the average daily throughput rate of the system. The experimental study shows that the proposed B-TS algorithm improves the system performance significantly.

While we solve the problem only from buffer size minimization perspective, the problem can be solved also in a multi-objective manner involving other objectives such as maximization of throughput rate, minimization of average work-in-process, and minimization of average system time. Moreover, these production oriented criteria can be integrated with some monetary criteria for profit maximization or cost minimization.

REFERENCES

Aksoy, K.H., Gupta, S.M., 2010. Near optimal buffer allocation in remanufacturing systems with N-policy. *Computers&Industrial Engineering*, 59, 496-508

Altiparmak, F., Dengiz, B., Bulgak, A.A., 2007. Buffer allocation and performance modeling in asynchronous assembly system operations: An artificial neural network metamodeling approach. *Applied Soft Computing*, 7, 946-956.

Amiri, M., Mohtashami, A., 2011. Buffer allocation in unreliable production lines based on design of experiments, simulation, and genetic algorithm. *International Journal of Advanced Manufacturing*, 1-13, doi 10.1007/s00170-011-3802-8.

Battini, D., Persona, A., Regattieri, A., 2009. Buffer size design linked to reliability performance: A simulative study. *Computers&Industrial Engineering*, 56(4), 1633-1641.

Bulgak, A.A., 2006. Analysis and design of split and merge unpaced assembly systems by metamodelling and stochastic search. *International Journal of Production Research*, 44 (18-19), 4067-4080.

Can, B., Heavey, C., 2011. Comparison of experimental designs for simulation-based symbolic regression of manufacturing systems. *Computers&Industrial Engineering*, 61(3), 447-462.

Can, B., Heavey, C., 2012. A comparison of genetic programming and artificial neural networks in metamodeling of discrete-event simulation models. *Computers&Operations Research*, 39(2), 424-436.

Cruz, F.R.B., Duarte, A.R., Van Woensel, T., 2008. Buffer allocation in general single-server queueing networks. *Computers&Operations Research,* 35(11), 3581-3598.

Demir, L., Tunali, S., Løkketangen, A., 2011. A tabu search approach for buffer allocation in production lines with unreliable machines. *Engineering Optimization*, 43(2), 213-231.

Demir L., Tunali, S., Eliiyi, D.T., 2012. An adaptive tabu search approach for buffer allocation problem in unreliable non-homogenous production lines. *Computers&Operations Research*, 39(7), 1477-1486.

Diamantidis, A.C., Papadopoulos, C.T., 2004. A dynamic programming algorithm for the buffer allocation problem in homogeneous asymptotically reliable serial production lines. *Mathematical Problems in Engineering*, 3, 209-223.

Dolgui, A., Eremeev, A., Kolokolov, A., Sigaev, V., 2002. A genetic algorithm for the allocation of buffer storage capacities in a production line with unreliable machines. *Journal of Mathematical Modeling and Algorithms*, 1, 89-104.

Dolgui A., Eremeev, A., Sigaev, V., 2007. HBBA: hybrid algorithm for buffer allocation in tandem production lines. *Journal of Intelligent Manufacturing*, 18, 411-420.

Gershwin, S.B., Schor, J.E., 2000. Efficient algorithms for buffer space allocation. *Annals of Operations Research*, 93, 117-144.

Glover, F., Taillard, E., Werra, D., 1993. A user's guide to tabu search. *Annals of Operations Research*. 41, 3-28

Glover, F., Laguna, M., 1997. *Tabu Search.* Dordrecht: Kluwer Academic Publishers.

Gurkan, G., 2000. Simulation optimization of buffer allocations in production lines with unreliable machines. *Annals of Operations Research*, 93, 177-216.

Helber, S., 2001. Cash-flow-oriented buffer allocation in stochastic flow lines. *International Journal of Production Research*, 39 (14), 3061-3083.

Jeong, K.-C., Kim, Y.-D., 2000. Heuristics for selecting machines and determining buffer capacities in assembly systems. *Computers&Industrial Engineering*, 38, 341-360.

Lutz, C.M., Davis, K.R., Sun, M., 1998. Determining buffer location and size in production lines using tabu search. *European Journal of Operational Research*, 106, 301-316.

Nahas, N., Ait-Kadi, D., Nourelfath, M., 2009. Selecting machines and buffers in unreliable series-parallel production lines. *International Journal of Production Research*, 47 (14), 3741-3774.

Nourelfath, M., Nahas, N., Ait-Kadi, D., 2005. Optimal design of series production lines with unreliable machines and finite buffers. *Journal of Quality in Maintenance Engineering*, 11(2), 121-138.

Qudeiri, J. A., Yamamoto, H., Ramli, R., Jamali, A., 2008. Genetic algorithm for buffer size and work station capacity in serial–parallel production lines. *Artificial Life and Robotics*, 12, 102-106.

Reeves, C.R., 1996. *Modern Heuristic Techniques in Modern Heuristic Search Methods*. Edited by Rayward-Smith, V.J., Osman, I.H., Reeves, C.R., Smith, G.D. England: John Wiley & Sons.

Sabuncuoglu, I., Erel, E., Kok, A.G., 2002. Analysis of assembly systems for interdeparture time variability and throughput. *IIE Transactions*, 34, 23-40.

Sabuncuoglu, I., Erel, E., Gocgun, Y., 2006. Analysis of serial production lines: characterization study and a new heuristic procedure for optimal buffer allocation. *International Journal of Production Research*, 44 (13), 2499-2523.

Seong, D., Chang, Y.S., Hong, Y., 2000. An algorithm for buffer allocation with linear resource constraints in a continuous-flow unreliable production line. *Asia-Pacific Journal of Operational Research*, 17, 169-180.

Shi, L., Men, S., 2003. Optimal buffer allocation in production lines. *IIE Transactions*, 35, 1-10.

Shi, C., Gershwin, S.B., 2009. An efficient buffer design algorithm for production line profit maximization. *International Journal of Production Economics*, 122, 725-740.

Spinellis, D., Papadopoulos, C.T., MacGregor Smith, J., 2000. Large Production Line Optimization Using Simulated Annealing. *International Journal of Production Research,* 38, 3, 509-541.

Spinellis, D., Papadopoulos, C. T., 2000a. A simulated annealing approach for buffer allocation in reliable production lines. *Annals of Operations Research*, 93, 373-384.

Spinellis, D., Papadopoulos, C.T., 2000b. Stochastic Algorithms for Buffer Allocation in Reliable Production Lines. *Mathematical Problems in Engineering*, 5, 441-458.

Tempelmeier, H., 2003. Practical considerations in the optimization of flow production systems. *International Journal of Production Research*, 41 (1), 149-170.

Yamashita, H., Altıok, T., 1998. Buffer capacity allocation for a desired throughput in production lines. *IIE Transactions*, 30, 883-891.