On Local Optima Distribution in Buffer Allocation Problem for Production Line with Unreliable Machines ⋆

Alexandre Dolgui * **Anton Eremeev** ** **Viatcheslav Sigaev** ***

* *IMT Atlantique, Nantes, France*
** *Sobolev Institute of Mathematics SB RAS, Novosibirsk, Russia (e-mail: eremeev@ofim.oscsbras.ru).*
*** *Avtomatika-Servis LLC, Omsk, Russia (e-mail: sigvs@yandex.ru).*

Abstract: In this paper, we consider a manufacturing flow-line organized as a series-parallel system of machines separated by finite buffers. The failure and repair times of machines are supposed to be exponentially distributed. The production rate of each machine is deterministic, and different machines may have different production rates. The buffer allocation problem consists in determining the buffer capacities with respect to a given optimality criterion, which depends on the average production rate of the line, the buffer acquisition and installation cost and the inventory cost. The tentative solutions are evaluated with an approximate method based on the Markov models aggregation approach. The computational experiments show better quality of solutions obtained by a genetic algorithm compared with the local descent and tabu-search algorithms. It is indicated that in many test problems several clusters of local optima can be found.

Keywords: Production line, Unreliable machines, Buffer allocation, Series-parallel network, Genetic algorithms, Local optima.

1. INTRODUCTION

Buffer capacity allocation problems arise in a wide range of manufacturing systems, such as transfer lines, flexible manufacturing or robotic assembly systems which are flow lines. Buffers separate any two consecutive machines. The parts are accumulated in a buffer when the machines downstream are less productive than machines upstream. Assume that machines can breakdown. When a breakdown occurs, the corresponding machine is not used in production for a random repair time which is independent on the number of failed machines. It is assumed that there is a sufficient number of raw parts at the input of the system and these parts are always available. The completed parts depart from the system immediately. The performance of the flow-line is measured in terms of the average production rate, i.e., the steady state average number of parts produced per unit of time.

In the literature, there are two types of publications. The first concerns only evaluation of the line performance for a given size of buffers. In the second, the buffer sizes are optimized. For example, Dallery and Gershwin (1992), Gershwin (1993), Heavey et al. (1993), Li and Meerkov (2009), and Tan and Gershwin (2009) proposed models to evaluate the performance of lines with unreliable machines and fixed sizes of buffers. Markov models and aggregation or decomposition techniques are often used to calculate steady state throughput or other performance indicators for these lines provided that the buffer capacities are given. Based on these models for performance analysis, in, e.g., Smith and Daskalaki (1988), So (1997), Gershwin and Schor (2000), and Shi and Gershwin (2009), the optimization for buffer capacity allocation was considered with respect to diverse optimality criteria and for different types of lines.

⋆ The research is supported by Russian Science Foundation grant 21-41-09017.

1.1 The Buffer Allocation Problem Formulation

In the present paper, we consider buffer allocation problem for line with a series-parallel network. Example of line with a series-parallel network shown in Fig. 1, where $M_1, \dots, M_7$ are machines and $B_0, \dots, B_5$ are buffers.

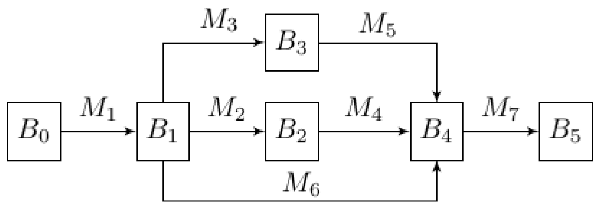

Fig. 1. Example of a line with a series-parallel network

We assume that a machine can be in an operational state or under repair. An operational machine may be blocked and temporarily stopped in case if there is no room in the downstream buffer. It may also be starved if there are no parts to process in the upstream buffer. Otherwise operational machines are working. In what follows, m denotes the number of machines in a production line. A working machine i, $i = 1, \dots, m$, is assumed to have a constant cycle time C_i and, then, the average production rate $u_i = 1/C_i$.

It is supposed that machines break down only when they are working. The times to fail and times to repair for each machine are assumed to be mutually independent and exponentially distributed random values. Let T_b^i denote the average time to fail, and let $\lambda_i = 1/T_b^i$ be the failure rate for working machine i, $i = 1, \ldots, m$. Similarly, let T_r^i and $\mu_i = 1/T_r^i$ denote respectively the time to repair and the repair rate for machine i. Under the above mentioned assumptions the system has the steady state mode (see e.g. (Sevast'yanov, 1962)), and performance of the system in this state is important for applications.

Let the buffers in the system be denoted by $B_1, \ldots, B_n$ and let h_j be the capacity of buffer B_j, which is to be decided. Denote the vector of decision variables as $H = (h_1, h_2, \ldots, h_n) \in Z_+^n$, where Z_+ is the set of non-negative integers.

Optimization criteria that was used is:

$$\max \phi(H) = T_{am} R(V(H)) - Q(H) - J(H), \qquad (1)$$

where

- T_{am} amortization time of the line (line life);
- $V(H)$ average production rate (steady state throughput);
- $R(V)$ revenue related to the production rate V;
- $J(H)$ cost of buffer configuration H;
- d_j maximal admissible capacity of buffer B_j, $j = 1, \ldots, n$;
- $Q(H) = c_1 q_1(H) + \ldots + c_n q_n(H)$ average steady state inventory cost, where $q_j(H)$ is the average steady state number of parts in buffer b_j, for $j = 1, \ldots, n$.

Function $\phi(H)$ has to be maximized, subject to the constraints $h_1 \le d_1$, $h_2 \le d_2$,..., $h_n \le d_n$. $R(V)$ and $J(H)$ are assumed to be given monotone non-decreasing real-valued functions. The cost function $J(H)$ may be non-linear to model some standard buffer capacities or penalize solutions where the total capacity of the buffers exceeds an upper bound. A non-linear revenue function $R(V)$ can model the law of diminishing returns, for example, it can reflect the effect of overproduction by switching from strictly increasing to constant at a certain threshold. A stepwise revenue function can be used to model zero revenue in case of an unacceptably low average production rate (see e.g. Section 3).

In general, the production rate with finite buffers is difficult to analyze precisely with the Markov models. Exact performance computation of a production rate of a line with more than two machines and one buffer is problematic due to exponential growth of the number of states. Therefore, most of the techniques employed for the analysis of such systems are in the form of analytic approximations and simulations. Analytical approximations are generally based on the two-machines-one-buffer Markov models, and either aggregation De Koster (1987) or decomposition approach Dallery et al. (1989); Gershwin (1987); Li (2005). Simulation models are more expensive computationally but applicable to a wider class of systems (Dolgui and Svirin (1995); Sörensen and Janssens (2004)).

In this paper, we use two-machines-one-buffer Markov model independently developed by Levin and Pasjko (1969), Dubois and Forestier (1982) and Coillard and Proth (1984). For each tentative buffer allocation decision, the production rate is evaluated via an aggregation algorithm (Dolgui (1993); Dolgui and Svirin (1995)), which is similar to the Terracol and David (1987) techniques. This aggregation approach appears to be sufficiently rapid for evaluation of tentative buffer allocations within the optimization algorithms.

The aggregation algorithm for production rate evaluation consists in recurrent replacement of two adjacent machines by a single machine. The parameters λ^*, μ^*, c^* of the resulting single machine are calculated from differential equations corresponding to the two-machines-one-buffer Markov model. After n steps of such aggregation procedure the system reduces to one virtual machine with parameters λ^*, μ^*, c^* and the estimate of the overall production rate $V(H)$ is given by $c^*\mu/(\lambda^* + \mu)$.

The buffer allocation problem is known to be NP hard as shown by Dolgui et al. (2013, 2018) and therefore it features some properties of the well-known combinarotial optimization problems, one of such properties is that often it is easy to find a locally optimal solution (computable in polynomial time w.r.t. the problem input size), although it is hard to find the global optimum (requires exponential time in the worst case).

1.2 Big Valley or Massif Central

In many combinatorial optimization problems, local optima tend to be clustered in a "big valley" (in the case of minimization problems) or "massif central" (in the case of maximization problems). This landscape structure has been observed e.g. in NK-landscapes by Kauffman and Levin (1987), in the traveling salesman problem by Boese et al. (1993); Hains et al. (2011), in the graph bisection by Boese et al. (1993), and flowshop scheduling by Reeves (1999). More precisely, the "big valley" or "massif central" is described by the following two statements (see e.g. Boese et al. (1993)):

(1) Values of the objective function of local optima tend to deteriorate with increasing distance to the global optimum.
(2) Local optima are located relatively close both to each other and to the global optimum (they are located in a ball, which is smaller than the whole search space by several orders of magnitude).

The presence of such structure partly explains good performance of genetic algorithms (GAs). If different local optima are found in the population of a GA and the new solution is built by means of a crossover operator, then the intuition suggests that such algorithm should have good chances to find the global optimum. This is supported by the theoretical analysis in the case of the Jump benchmark of Dang et al. (2018), and by the experimental studies, e.g. of Hains et al. (2011). In this respect, identification of the "big valley" or "massif central" structure, or the absence of such structure, is of great practical interest.

1.3 Contribution of the Paper

We study the distribution of local optima for buffer allocation problem for lines with a series-parallel structure. The location of local optima for each task is evaluated by the multi-start of a local search alghoritm. After that we verify the massif central conjecture. On basis of obtained results, the behavior of genetic algorithm and tabu search algorithm on each task is substantiated.

2. INVESTIGATION OF LOCAL OPTIMA PROPERTIES

The distance to global optimum in our case, we will calculate in the metric l_1. To verify the big valley conjecture, a method was developed for finding the number of integer points in a ball of a given radius in the metric l_1 at the intersection with a parallelepiped whose faces are parallel to coordinate planes. The method was obtained by reducing the problem to a combinatorial formulation, already considered earlier for the case of non-negative integer points, using generating functions **?**.

2.1 Tasks Used In Computational Experiment

For computational experiments we use three series of problems:

Series	Number of Lines	Number of Machines
AS	8	4 – 14
BN	10	5
VP	4	5

Table 1. List of series

The AS series consists of tasks created from lines 1,2,6,7,8 from Ancelin and Semery (1987) with real data from produced by Renault. A distinctive feature of the 7 and 8 lines is the presence parallel sections (see figures 2, 3). Line parameters 1,2,6,7,8 are given in tables 2-6.

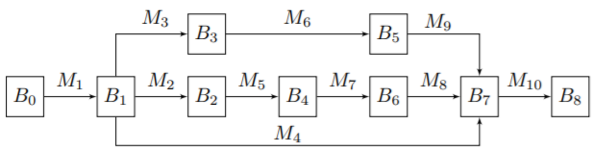

Fig. 2. Line structure AS7

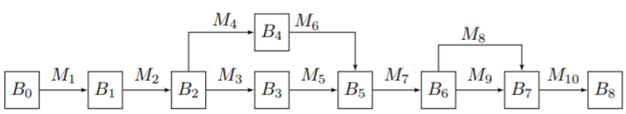

Fig. 3. Line structure AS8

buffers		machines			
i	d_i	j	T_j^{O}	T_j^{B}	U_i
1	20	1	244.2	150	10
2	17	2	255.3	300	10
3	38	3	176	75	10
4	48	4	184	600	10
		5	192	450	10

Table 2. Task parameters *AS1*

bn5.1-bn5.10 series consists of 10 lines with 5 machines in each from Dolgui et al. (2000). There are two bottlenecks

Buffers		Machines			
i	d_i	j	T_j^{O}	T_j^{B}	U_i
1	0	1	10000	440	22
2	50	2	20000	440	23
3	20	3	5000	430	22
4	50	4	40000	520	23
5	0	5	30000	430	24
6	80	6	2442	440	22
7	20	7	1840	520	23
8	100	8	1680	430	21
9	100	9	2208	920	24

Table 3. Task parameters *AS2*

Buffers		Machines			
i	d_i	j	T_j^{O}	T_j^{B}	U_i
1	60	1	29880	22000	385
2	60	2	29880	22000	426
3	50	3	876000	22300	330
4	70	4	29880	22000	372
5	60	5	33250	27500	316
6	80	6	144000	8500	340
7	45	7	102300	74000	340
8	25	8	113300	7200	340
9	35	9	540000	60000	380
10	80	10	538800	349000	350
11	40	11	5064000	73700	400
12	45	12	468000	306000	400
13	65	13	1032000	54000	319
		14	45600	31120	319

Table 4. Task parameters *AS6*

Buffers		Machines			
i	d_i	j	T_j^{O}	T_j^{B}	U_i
1	15	1	50000	12000	1000
2	10	2	48000	2000	3450
3	15	3	55000	9000	2780
4	10	4	39000	6000	3030
5	25	5	75000	10000	3333
6	10	6	59000	11000	2560
7	10	7	28000	8000	3030
		8	35000	8000	3125
		9	65000	35000	2174
		10	20000	4000	800

Table 5. Task parameters *AS7*

Buffers		Machines			
i	d_i	j	T_j^{O}	T_j^{B}	U_i
1	1300	1	87000	27000	23
2	200	2	77000	22000	27
3	0	3	580000	18000	38
4	0	4	410000	12500	30
5	0	5	580000	18000	38
6	2000	6	410000	12500	30
7	0	7	725000	21000	20
		8	550000	14000	40
		9	430000	24000	43
		10	270000	22000	33

Table 6. Task parameters *AS8*

in the problems of this series. Under bottleneck is understood as a line section of two relatively slow machines and a buffer between them.

Series vp6.9-vp6.10 and vp7.9-vp7.10 Vouros and Papadopoulos (1998) are composed of problems on lines from

5 consecutive machines from Vouros and Papadopoulos (1998).

2.2 Computational Experiment

A series of experiments were carried out on existing tasks for determination of the structure of local optima. With a one-time running the local descend algorithm LDA solution H whose elements h_i are chosen with uniform distribution between 1 and d_i improved to a local optimum. This procedure was repeated 300 times to create the necessary sample size (Boese et al. (1993)). Based on the results obtained, it was calculated number of admissible solutions $\mid \Omega \mid$ in balls containing all found local optima. In this case, distance from best found local optimum to worst, and beyond center - best local optimum found.

In table 7-10 the column $V1$ contains cardinality $\mid \Omega \mid$ for balls containing local optima, the column $V2$ contains cardinality of entire space solutions for each problem, and the column $V1/V2$ is their ratio. As can be seen from tables, first part of the thesis, that local optima are located relatively close both to each other and to global optimum, in our case, it was not performed and mainly in tasks used the optima are strongly scattered throughout the solution space.

№	$V1$	$V2$	$V1/V2$
1	2,84E+05	5,74E+05	0,49
2	8,88E+11	9,48E+11	0,94
3	8,60E+01	4,85E+03	0,02
4	2,34E+22	2,89E+22	0,81
5	4,17E+07	9,75E+07	0,43
6	1,00E+00	5,23E+08	0,00

Table 7. Results of running a local descend multiple times for a series as.1 - as.6

№	$V1$	$V2$	$V1/V2$
9	7,97E+03	1,00E+04	0,80
10	7,55E+03	1,46E+04	0,52

Table 8. Results of running a local descend multiple times for a series vp6.9 - vp6.10

№	$V1$	$V2$	$V1/V2$
9	6,50E+03	1,00E+04	0,65
10	9,95E+03	1,46E+04	0,68

Table 9. Results of running a local descend multiple times for a series vp7.9 - vp7.10

Also considered was a part of the thesis about tendency to worsen value of objective function of local optima with increasing distance to global optimum. To do this, for all considered problems, correlation was determined $\rho(\varphi(\xi), r(\xi, \xi^*))$ objective function values local optima $\varphi(\xi)$ and distance to global optimum $r(\xi, \xi^*)$. As shown by experiments on proposed problems, there is a negative correlation ρ, and all values correlations are significantly different from 0 with a confidence level of 95% (tables 11–12).

An interesting fact is that that in problems as.6, bn5.1 entire set of local optima splits into clusters, for each of which a part of the thesis about tendency to deterioration of the value of the objective function of local optima with increasing distance to global optimum.

№	$V1$	$V2$	$V1/V2$
1	1,44E+05	1,94E+05	0,74
2	9,96E+03	1,94E+05	0,05
3	9,07E+04	1,94E+05	0,47
4	2,06E+04	1,94E+05	0,11
5	6,95E+04	1,94E+05	0,36
6	6,94E+04	1,94E+05	0,36
7	2,45E+04	1,94E+05	0,13
8	3,42E+03	1,94E+05	0,02
9	6,28E+04	1,94E+05	0,32
10	1,00E+00	1,94E+05	0,00

Table 10. Results of running a local descend multiple times for a series bn5.1 - bn5.10

№	$\rho(r(\xi, \xi*), \varphi(\xi))$
Series as.1 - as.5	
1	-0,87078206
2	-0,437208613
7	-0,547348483
8	-0,943725905
Series vp6.9 - bn6.10	
9	-0,833583818
10	-0,778237651

Table 11. Results of running a local descend multiple times for a series *as* and *vp*6

№	$\rho(r(\xi, \xi*), \varphi(\xi))$
Series bn5.1 - bn5.9	
1	-0,706451573
2	-0,872602935
3	-0,914939714
4	-0,999969666
5	-0,972555479
6	-0,729655068
7	-0,973827419
8	-0,846807033
9	-0,884916669
Series vp7.9 - bn7.10	
9	-0,907792301
10	-0,889724863

Table 12. Correlation results for series *bn* and *vp*7

Figures 4 and 5 show diagrams of local optima for problems as.6, bn5.1, where ordinate shows the value of objective function of local optimum $\varphi(H)$, and abscissa is distance in metric l_1 to chosen solution.

Clustering effect is especially well manifested in tasks with parallel sections of line. This effect can be justified by fact that there are several different paths in lines with a parallel structure. from start buffer to end buffer. Thus, if you create two lines that are identical in graph structure, such that one line will have relatively large buffers along same path, and other - otherwise, it is possible to achieve two solutions close in value to the objective function, but located on "enough" great distance from each other.

Another reason for clustering effect is the property line symmetry. This property is set to Levin and Pasjko (1969) for a serial line, consisting of two machine and a buffer

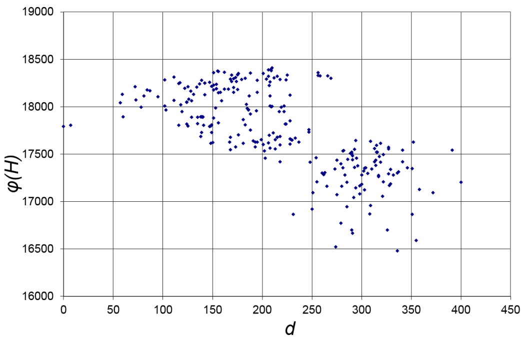

Fig. 4. Structure of local optima of problem as6.

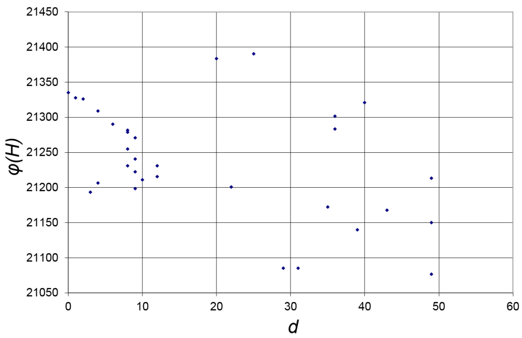

Fig. 5. Structure of local optima of problem bn5.1.

between them. If we swap the parameters of first and second machines so that that input buffer of line will become output, and output - input, then line performance will not change.

Consider the example of serial line t.1 from three machiness and two buffers between them with following parameters: $T_i^{\mathrm{O}} = T_i^{\mathrm{B}} = 1, i = 1, \ldots, 3$; $U_1 = 1, U_2 = 0.5$, $U_3 = 1$; $d_j = 4, c_j = 0, j = 1, \ldots, 2$; $T_{am} = 7000$; $J(H) = 50 \cdot (h_1 + h_2)$; if $V(H) < 2570$ then $R(V(H)) = 0.9 \cdot V(H)$, otherwise $R(V(H)) = 2570$.

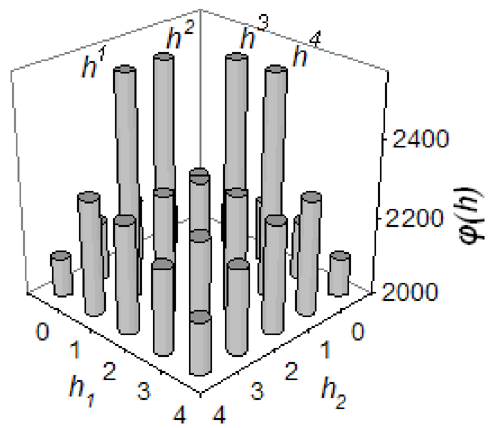

Fig. 6. Values of the objective function on the set D for problem t.1

Set of global optima (see Fig. 6) of this example splits into two clusters. In first cluster contains solutions $H^1 = (1, 2)$ and $H^2 = (1, 3)$, and the second is the solutions $H^3 = (2, 1)$ and $H^4 = (3, 1)$. Clustering optima in considered example is a consequence of symmetry and singularities aggregation algorithm from Dolgui (1993) used in calculating average line performance.

In general case, for lines with $n \geq 3$ buffers, effect symmetry can be observed in several internal areas lines. This leads to fact that with this aggregation algorithm set of local optima is divided into several clusters.

If a buffer allocation problem has local optima with different values of objective function, then it can become difficult for local descend algorithm. Sequence of points generated by this algorithm and falling into one of a clusters, as as a rule, remains in it until end of the calculations. For a tabu-search algorithm, transition between clusters is unlikely, which also makes such tasks are difficult for him. Best performance for considered problems, compared with the other two algorithms, showed the genetic algorithm GA and GA with local optimization heuristic from Dolgui et al. (2007). As experiments have shown, the population of the genetic algorithm contains individuals that correspond to solutions from different clusters. Using of selection, crossover and mutation operators allows maintaining the diversity of population. There is a competitive struggle between different clusters for their representative in the population at each step of GA.

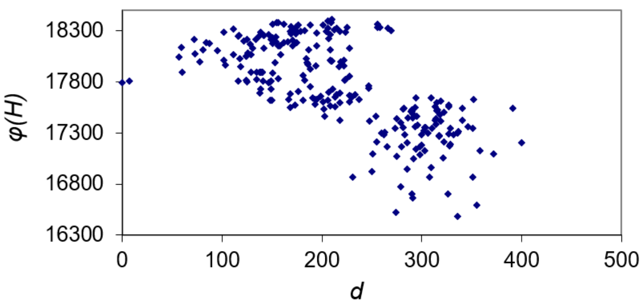

Fig. 7. Local optimums obtained by local descent multistart

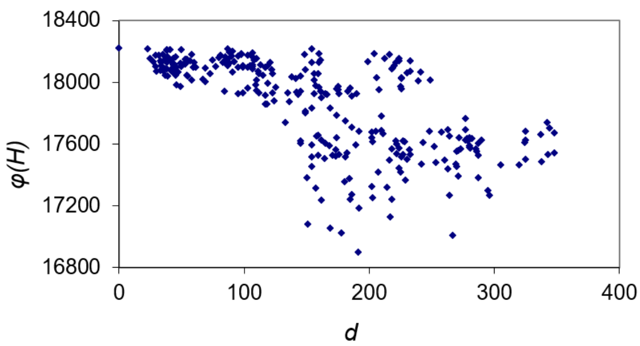

Fig. 8. Final population of GA

Task AS4 can serve as an illustrative example. Figure 7 shows structure of local optima obtained by running the local descent algorithm multistart. Figures 8 and 9 show final populations of GAs and GAs with local optimization heuristic, which show the cluster structure of final population. Figure 9 shows the work of the local heuristic, its use allows leaving only the best individuals in each cluster.

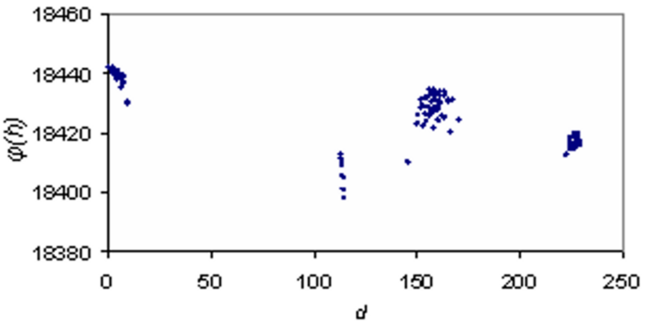

Fig. 9. Final population of GA with local optimization heuristic

3. CONCLUSION

(1) Cluster structure of local optima is established of some tasks having parallel parts in structure.
(2) Reasons for the emergence of clusters of local optima are determined..

REFERENCES

Ancelin, B. and Semery, A. (1987). Calcul de la productivité d'une ligne integrée de fabrication. *RAIRO Automatiq., Productiq. Informatiq. Industrielle*, 21, 209–238.

Boese, K., Kahng, A., and Muddu, S. (1993). On the big valley and adaptive multi-start for discrete global optimizations. *Technical report, UCLA CS Department, TR-930015.*

Coillard, P. and Proth, J. (1984). Effet des stocks tampons dans une fabrication en ligne. *Revue belge de Statistique, d'Informatique et de Recherche Opérationnelle*, 24 (2), 3–27.

Dallery, Y., David, R., and Xie, X. (1989). Approximate analysis of transfer lines with unreliable machines and finite buffers. *IEEE Transactions on Automatic Control*, 34, 943–953.

Dallery, Y. and Gershwin, S. (1992). Manufacturing flow line systems: a review of models and analytical results. *Queueing Syst.*, 12(1-2), 3–94.

Dang, D.C., Friedrich, T., Koetzing, T., Krejca, M., Lehre, P., Oliveto, P., Sudholt, D., and Sutton, A. (2018). Escaping local optima using crossover with emergent diversity. *IEEE Transactions on Evolutionary Computation*, 22(3), 484 – 497. doi:10.1109/TEVC.2017.2724201.

De Koster, M. (1987). Estimation of line efficiency by aggregation. *International Journal of Production Research*, 25, 615–626.

Dolgui, A. (1993). Analyse de performances d'un atelier de production discontinue: méthode et logiciel. *Research Report INRIA*, 1949.

Dolgui, A., Eremeev, A., Kolokolov, A., and Sigaev, V. (2000). A genetic algorithm for buffer allocation in production line with unreliable machines. In *Proceedings of the International Workshop Discrete Optimization Methods in Scheduling and Computer-Aided Design*, 26–31.

Dolgui, A., Eremeev, A., and Sigaev, V. (2007). Hbba: hybrid algorithm for buffer allocation in tandem production lines. *Journal of Intelligent Manufacturing*, 18(3), 411–420.

Dolgui, A. and Svirin, Y. (1995). Models of evaluation of probabilistic productivity of automated technological complexes. *Vesti Akademii Navuk Belarusi: phisikatechnichnie navuki*, 1, 59–67.

Dolgui, A., Eremeev, A., Kovalyov, M., and Sigaev, V. (2018). Complexity of Bi-objective Buffer Allocation Problem in Systems with Simple Structure. In *Optimization Problems and Their Applications*, volume 871 of *Communications in Computer and Information Science*, 278–287. Springer. URL `https://hal.archives-ouvertes.fr/hal-01977539`.

Dolgui, A., Eremeev, A., Kovalyov, M.Y., and Sigaev, V. (2013). Complexity of Buffer Capacity Allocation Problems for Production Lines with Unreliable Machines. *Journal of Mathematical Modelling and Algorithms in Operations Research*, Volume 12(Issue 2), pp 155–165. doi:10.1007/s10852-012-9199-z. URL `https://hal-emse.ccsd.cnrs.fr/emse-00881377`.

Dubois, D. and Forestier, J. (1982). Productive et encours moyens d'un ensemble de deux machines separees par une zone de stockage. *RAIRO Automatique*, 16(12), 105–132.

Gershwin, S. (1987). An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking. *Operations Research*, 35 (2), 291–305.

Gershwin, S. (1993). *Manufacturing Systems Engineering.* Prentice Hall.

Gershwin, S. and Schor, J. (2000). Efficient algorithms for buffer space allocation. annals of operations research. *Annals of Operations Research*, 93, 117–144.

Hains, D., Whitley, L., and Howe, A. (2011). Revisiting the big valley search space structure in the tsp. *Oper. Res. Soc. 62*, 305–312. doi: https://doi.org/10.1057/jors.2010.116.

Heavey, C., Papadopoulos, H., and Browne, J. (1993). The throughput rate of multistation unreliable production lines. *Europ. J. Oper. Res.*, 68, 69—-89.

Kauffman, S. and Levin, S. (1987). Towards a general theory of adaptive walks on rugged landscapes. *Journal of Theoretical Biology*, 128(1), 11–45. doi: https://doi.org/10.1016/S0022-5193(87)80029-2.

Levin, A. and Pasjko, N. (1969). Calculating the output of transfer lines. *Stanki i Instrument*, (8), 8–10.

Li, J. (2005). Overlapping decomposition: a system-theoretic method for modeling and analysis of complex manufacturing systems. *IEEE Transactions on Automation Science and Engineering*, 2 (1), 40–53.

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

Reeves, C. (1999). Landscapes, operators and heuristic search. *Annals of Operations Research*, 86(0), 473–490. doi:10.1023/A:1018983524911. URL `https://ideas.repec.org/a/spr/annopr/v86y1999i0p473-49`

Shi, C. and Gershwin, S. (2009). An efficient buffer design algorithm for production line profit maximization original research. *International Journal of Production Economics*, 122 (2), 725–740.

Smith, J. and Daskalaki, S. (1988). Buffer space-allocation in automated assembly lines. *Operations Research*, 36, 343–358.

So, K. (1997). Optimal buffer allocation strategy for minimizing work-in-process inventory in unpaced production lines. *lIE Transactions*, 29, 81–88.

Sörensen, K. and Janssens, G. (2004). A petri net model of a continuous flow transfer line with unreliable machines. *European Journal of Operational Research*, 152, 248–262.

Tan, B. and Gershwin, S. (2009). Analysis of a general markovian two-stage continuous-flow production system with a finite buffer. *International Journal of Production Economics*, 120 (2), 327–339.

Terracol, C. and David, R. (1987). Performance d'une ligne composée de machines et de stocks intermédiaires. *RAIRO Automatiq., Productiq. Informatiq. Industrielle*, 21, 239–262.

Vouros, G. and Papadopoulos, H. (1998). Buffer allocation in unreliable production lines using a knowledge based system. *Computers Ops. Res.*, 25(12), 883–891.