Fitness Landscapes of Buffer Allocation Problem for Production Lines with Unreliable Machines

Alexandre Dolgui[a], Anton V. Eremeev[b], Vyatcheslav S. Sigaev[c]

[a] *IMT Atlantique, LS2N-CNRS, Nantes, France*
[b] *Sobolev Institute of mathematics, Omsk Department, Omsk, Russia*
[c] *Avtomatika-Servis LLC, Omsk, Russia*

Abstract

We study the structural properties of the buffer allocation problem from the fitness landscape perspective. We consider manufacturing flow lines with series-parallel network structure. The machines are supposed to be unreliable, their time to failure and repair time are exponentially distributed. Tentative solutions are evaluated by means of an approximate method based on the Markov models aggregation. We carry out computational experiments with local search and genetic algorithms in order to evaluate the fitness landscape properties of previously published instances and their modifications. It turns out that the so-called 'massif central' or 'big valley' structure of the fitness landscape is present but only partially: The fitness of local optima is negatively correlated with the distance to the best found solution, yet the set of local optima cannot be encompassed by a ball of relatively small size with respect to the size of solution space. Moreover, we show that in many problem instances, several clusters of local optima can be identified. The performance of genetic algorithms is discussed with respect to population clustering and the permanent usage of crossover is recommended.

Keywords: Production line, unreliable machines, buffer allocation, series-parallel network, genetic algorithms, local optima.
PACS: 02.60.Pn
2000 MSC: 90C27, 90C59, 90B05

1. Introduction

Buffer capacity allocation problems arise in a wide range of flow line manufacturing systems, such as transfer lines, flexible manufacturing or robotic

assembly systems. The parts are accumulated in the intermediate buffers when the downstream machines are less productive than the upstream machines. It is assumed that machines can break down and then go through repair. When a breakdown occurs, the corresponding machine is not used in production for a random repair time, which is independent of the total number of machines under repair. We assume that there is a sufficient number of raw parts at the input buffer of the line and the finished parts depart from the system immediately. One of the key performance measures of a flow-line is the average production rate, i.e., the expected number of parts produced per unit of time in the steady state regime. We also consider the inventory cost and the buffers costs.

Evaluation of the manufacturing flow-line performance for given sizes of buffers is studied by Coillard and Proth (1984), Dallery and Gershwin (1992), Heavey et al. (1993), Tan and Gershwin (2009), Dolgui and Proth (2006) and other authors. Markov models (see the recent review of Papadopoulos et al. (2018)) and aggregation or decomposition techniques (see e.g. (Dallery et al., 1988, 1989; Dolgui, 1993; Gershwin, 1993; Li and Meerkov, 2009)) are often used to calculate the steady state throughput or other performance indicators for the lines. Here we use the same assumptions as made by Coillard and Proth (1984) (see the details in Subsection 2.1 below) and extend that line of research.

Comprehensive surveys on buffer allocation in production lines may be found in (Demir et al., 2014; Weiss et al., 2018). Buffer capacity allocation using Markov models was studied by Smith and Daskalaki (1988), So (1997), Gershwin and Schor (2000), Shi and Gershwin (2009), Kassoul et al. (2021) and other authors. A promising approach, based on artificial neural networks for modelling the throughput, has been recently proposed in (Dieleman et al., 2023).

A large body of literature is devoted to buffer size optimization using the simulation-optimization approach. Usually, in case of simulation-optimization it is necessary to iterate between optimization and simulation procedures, see e.g. (Fu, 2002). However each call to a simulation procedure usually requires much more CPU time compared to the evaluation by means of a Markov model. The crucial problem in this framework is how to allocate the computational resource over the tentative solutions, given the current information about the problem. One of the best known methods developed in this framework is the *optimal computing budget allocation* (Chen and Lee, 2011). An alternative approach in production lines simulation-optimization

uses the *finite perturbation analysis* (Suri, 1989) that provides a solution to the design problem using a single long simulation of the system, which significantly reduces the computation time. During this simulation, the algorithm obtains updated values of the considered performance measure at perturbed solutions. The updated values of the performance measure are used to define the new search direction. Successful application of this approach may be found e.g. in the genetic algorithm of Kassoul et al. (2022). A new version of simulation-optimization approach may be found in (Alfieri et al., 2020), where the buffer allocation problem is approximated by means of a large-scale mathematical programming model. The approximation consists in modelling the queues with temporal lags which allows to avoid the loop between simulation and optimization modules. The buffer allocation methods, based on such mathematical programming models, are beyond the scope of the present paper, which is aimed at further development of the metaheuristic methods for buffer allocation, where the properties of the so-called *fitness landscapes* play an important role (Schiavinotto and Stützle, 2007; Stadler, 2002). Informally speaking, the fitness landscape describes the structure of the neighborhood system and the fitness function, representing the optimization criterion and problem constraints. This allows to focus the study on the set of local optima of a given problem instance with respect to the given neighborhood system. The constraints may be incorporated into the fitness function as penalty terms.

The production line design and buffers usage policy may significantly influence performance characteristics of a line. In particular, the production rate and expected inventory levels of the lines described above may significantly differ from those in production systems with rework loops (Biller et al., 2010), merging and splitting topologies (Smith and Daskalaki, 1988), production lines operating under an echelon buffer policy, and CONWIP policies (Liberopoulos, 2020). These cases are beyond the scope of our paper. Even the production lines, designed as we assume here, but with the information blocking policy (Buzacott and Shantikumar, 1993) or random processing time assumptions (Vouros and Papadopoulos, 1998), require separate studies, although such cases appear to be very similar in design.

In general, development and refining of buffer allocation algorithms is crucial for researchers of the field. For example, introduction of optimization software in PSA Peugeot Citroen for buffer allocation in production lines and other design decisions (Patchong et al., 2003) caused significant financial results. The optimization tools provide a manager with the decision

support which takes into account multiple constraints and objectives so that the insights about the most appropriate optimization algorithms and their settings allow to go further than general managerial insights about buffer allocation structures. The former is the main motivation of our research and is complementary to the later.

1.1. Contribution of the paper

The computational experiments with local search and genetic algorithms allowed us to evaluate the fitness landscape properties of buffer allocation instances, including the ones based on the real-life data from Renault production. It turned out that the well-known 'massif central' or 'big valley' structure (Boese et al., 1993) of the fitness landscape is present but only partially: Fitness of the local optima is negatively correlated with the distance to the best found solution, yet the set of local optima can not be encompassed by a ball of relatively small size with respect to the size of solution space. Moreover, we show that in many problem instances, several clusters of local optima can be identified. We also discuss the causes for problem symmetries and their impact on local optima clustering. This analysis of the fitness landscape leads to a better understanding of its geometric features and gives insights about the most appropriate optimization algorithms and their settings. In particular, in case of population clustering in genetic algorithm for buffer allocation problem, permanent usage of crossover operator is recommended.

To the best of our knowledge, this paper is the first detailed analysis of fitness landscapes of buffers allocation problems in the literature. Previous experimental studies of the fitness landscapes were mostly focused on combinatorial optimization problems, such as traveling salesman problem (Hains et al., 2011), the graph bisection (Boese et al., 1993), flowshop scheduling (Reeves, 1999), and recently have been extended to machine learning and some other areas (Thomson et al., 2017; Rodrigues et al., 2022).

This paper is an extended version of the conference publication (Dolgui et al., 2022). In this version, a detailed analysis of performance of genetic algorithm with respect to clustering of local optima is added. It allows us to recommend permanent use of crossover when population clustering emerges. More information on fitness landscapes of buffer allocation problem is provided in tables and figures, and the literature survey is significantly extended.

2. Preliminaries

2.1. The Buffer Allocation Problem

In this paper, we consider the buffer allocation problem for lines with a series-parallel structure. An example of such a line with a series-parallel network of equipment is shown in Fig. 1, where $M_1, \ldots, M_7$ are machines and $B_0, \ldots, B_5$ are buffers.

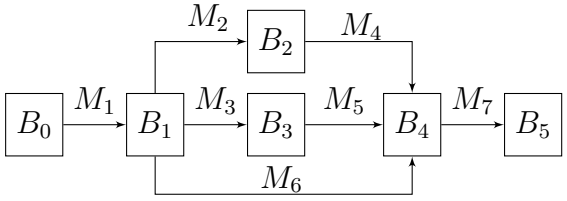

Figure 1: Example of a line with a series-parallel structure

We assume that a machine can be either operational or under repair. An operational machine may be *blocked* in case the downstream buffer is full. It may also be *starved* if there are no parts in the upstream buffer. Otherwise operational machines are working. In what follows, m denotes the number of machines in the system. A working machine i, $i = 1, \ldots, m$, is assumed to have a constant cycle time C_i, so its average production rate is $U_i = 1/C_i$.

It is supposed that machines may break down only when they are working. The time to fail and time to repair for each machine are assumed to be random values with exponential distributions. Let T_b^i denote the average time till failure, and let $\lambda_i = 1/T_b^i$ be the failure rate for a machine i, $i = 1, ..., m$, if this machine is working. 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, conditioned that this machine is under repair. Given our assumptions, the system has the steady state mode (see e.g. Sevast'yanov (1962)). The performance of the system in this mode is the most important for applications.

Let h_j be the capacity of buffer B_j, $j = 1, \ldots, n$. 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.

The optimization criterion used in this paper is the same as in Dolgui et al. (2002):

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

where

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

The function $\phi(H)$ has to be maximized, subject to the constraints $h_1 \leq d_1$, $h_2 \leq d_2, \ldots$, $h_n \leq d_n$, bounding the admissible buffer size. Functions $R(V)$ and $J(H)$ are assumed to be monotone and non-decreasing. $J(H)$ may be a linear function, or e.g. a step-function to model some standard buffer capacities, or may be a penalty function, imposing a penaly on solutions where the total capacity of all buffers exceeds some upper bound.

Although production lines without buffers or infinite buffers may be exactly described analytically, the lines with finite buffers are difficult to analyse precisely (Buzacott and Shantikumar, 1993). In particular, exact computation of the production rate and inventory levels in a line with more than two serial machines is problematic due to the exponential growth of the number of states in the corresponding Markov model. Therefore, most of the techniques developed for the analysis of such systems are based on analytical approximations or simulations. The analytical approximations are usually based on the two-machine Markov models, and either aggregation (De Koster, 1987) or decomposition (Dallery et al., 1989; Gershwin, 1987; Li, 2005; Diamantidis et al., 2020; Alaouchiche et al., 2023) algorithms.

For analytical approximation, in this paper, we use the two-machine Markov model, independently developed by Levin and Pasjko (1969), Dubois and Forestier (1982) and Coillard and Proth (1984). Given a tentative buffer allocation, the production rate is evaluated using the aggregation procedure (Dolgui, 1993) which is similar to the procedures of Terracol and David (1987) and Ancelin and Semery (1987). The aggregation algorithm consists in recursive replacement of two adjacent machines by a single machine. The parameters λ^*, μ^*, c^* of each emerging machine are calculated from Kolmogorov equations corresponding to the Markov model that describes

two machines in series or from a simple approximation that describes two machines in parallel. After n iterations of such aggregation procedure the system reduces to a single machine with parameters λ^*, μ^*, c^* and the estimate of the overall production rate $V(H)$ is given by $c^*\mu/(\lambda^* + \mu)$. The steady state inventory level q_i is found when the two-machine Markov model is applied to eliminate a buffer i during the reduction procedure Dolgui et al. (2002). This aggregation approach is sufficiently rapid for evaluation of tentative buffer allocations in the optimization algorithms. Computational experiments of Ancelin and Semery (1987) and Dolgui and Svirin (1995) show that the production rate $V(H)$ found by this method is within 95% confidence interval of the simulation results in most of the cases. The estimate for inventory level is rigorously established for the case of two-machine lines (Dolgui et al., 2002), however for longer lines the precision of our aggregation approach in computing $q_1(H), \ldots, q_n(H)$ may have a methodological bias.

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 combinatorial 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). Another important property is that high quality (and sometimes optimal) solutions may be found by means of problem decomposition into smaller subproblems and a proper way of solving and combining the solutions from these subproblems. Recent applications of the latter approach may be found in (Xi et al., 2022) and references therein. However in this paper we focus only on the first approach.

2.2. Fitness Landscapes

Many heuristic search methods for hard combinatorial optimization problems are based on the local search. Typically, the search is focused on a *neighbourhood* $\mathcal{N}_x$ of the current point x in the search space, and various strategies can be used for deciding on whether or not to move to one of these neighbours, and if so, to which. In practice, such methods quickly converge to a local optimum, and the search must either begin again by starting from a new point, or use some metaheuristic to guide the search into new areas.

It may be helpful to understand the structure of the so-called *fitness landscape* of a given instance of the combinatorial optimization problem in order to choose and to tune the metaheuristics for overcoming the low-quality

local optima and finding global solutions. Formally, a fitness landscape in the case of an optimization problem $\max\{\phi(x) : x \in X\}$ may be defined by three ingredients (Stadler, 2002):

1. A search space X,
2. a family of neighborhoods $\{\mathcal{N}_x\}$, where $\mathcal{N}_x \subset X$ is defined for all $x \in X$,
3. an objective function $\phi : X \to \mathbf{R}$, (also called *fitness function*).

In the present paper, we assume that a neighborhood $\mathcal{N}_x$, consists of all buffer allocations, which differ by one unit in one of the buffers from the given solution x. This definition is appropriate because the local search algorithms considered here always stop in a solution, which can not be improved by a move in such a neighborhood. For other versions of buffer allocation problems and other local search algorithms it may be more appropriate to use different definitions of a neighborhood, e.g. in the knapsack-type problem (Demir et al., 2012), the neighborhood is defined by moving a unit of storage from one buffer to another one.

Theoretical investigations of the capacity of metaheuristics to overcome the low-quality local optima have been recently shed some light on this issue (see e.g. Oliveto et al. (2018); Dang et al. (2021) and references therein), however in order to apply these findings to practical optimization problems one needs to perform a thorough analysis of the search space, which is a costly procedure. An alternative approach consists in statistical analysis of the outcomes of multiple restarts of a local search method, which may help to draw some qualitative conclusions on how to design a metaheuristic for the problem at hand (Boese et al., 1993; Thomson et al., 2017; Reeves and Eremeev, 2004). In this paper, we use the approach of Boese et al. (1993) to study the fitness landscape of buffer allocation problem and then perform computational experiments with different versions of genetic algorithm.

2.3. Genetic Algorithms

Let us briefly describe the two versions of genetic algorithm (GA) proposed in (Dolgui et al., 2002) for buffer allocation and the local optimization heuristics used in it. The GAs in general belong to a family of heuristic random search methods and based on the analogy of biological evolution. The GAs process a population of individuals by means of random operators that model mutation and crossover. Each individual of a population corresponds to a tentative solution, which is coded by a fixed length genotype string G, consisting of symbols, called genes. We consider GAs where after

crossover and mutation a new individual is processed by a local optimization heuristic, and only then it may be added into a population. The GAs of this type are also called *memetic algorithms* (Neri et al., 2012), These algorithms could be also classified as hybrid genetic algorithms with Lamarckian learning, however, since we borrowed them from (Dolgui et al., 2002), where they are called GAs, so keep calling them GAs to avoid a confusion.

Genotypes of the initial population are randomly generated according to some a priori defined probability distribution. All individuals are evaluated using the performance evaluation heuristic method from (Dolgui, 1993). The objective function is usually called *fitness function* in GA literature. The steady-state reproductive strategy is used here. According to this strategy, in each iteration, a couple of new individuals is generated. The new genotypes replace a couple of 'unpromising' individuals chosen in the population by some heuristic rule (thus the population size M remains constant during the run of a GA). For the stopping criterion, we use a given number of GA iterations t_{max} or a given computation time. The best solution found is returned as a result when a GA terminates.

The construction of a new pair of genotypes (offspring) starts with choosing a pair of parents from the current population by means of a probabilistic selection operator. Then the crossover operator replaces some genes of one parent with the corresponding genes of the other. After crossover the strings undergo mutation, where a randomly chosen subset of genes is randomly altered.

Three versions of the GA, the GA_0, GA_1 and GA_2 are considered. The difference between these algorithms is that in the GA_0, an individual is inserted into a population without any local optimization, while the GA_1 and GA_2 use the local optimization procedures respectively LO_1 or LO_2, described below.

In this paper, we consider an encoding of buffer allocation, which consists of n genes, and each gene contains the size of the corresponding buffer. Here we denote the genotypes of current population P by $G^{(1)}, G^{(2)}, \ldots, G^{(M)}$, and $G_i^{(j)}$ is the value of gene i (i.e. the size of buffer i) in genotype of individual j.

The genotypes of the initial population are randomly constructed with the uniform distribution of each gene $G_i^{(j)}$ over the set $0, 1, \ldots, d_i$, $i = 1, .., n$. After that in GA_1 and GA_2 the genotypes are processed by the same local optimization heuristic as the one used for local optimization of solutions before they are added into the population in each iteration of the GA (by iteration we mean construction of a pair of new genotypes from two parent genotypes).

Local optimization heuristic $LO_1(G)$ aims to improve a given individual G through incrementing or decrementing the size of each buffer by one. It makes at most n first-improving steps of local search using the ball of radius 1 as a neighbourhood. The other local search heuristic LO_2 has the following outline.

Local optimization heuristic $LO_2(G)$

1. Set $f := \phi(G)$
2. For all i from 1 to n do:
   2.1 Set $h := G_i$, $\delta := 1$
   2.2 Denote $H(\gamma) := (G_1, \ldots, G_{i-1}, h + \gamma, G_{i+1}, \ldots, G_n)$
   2.3 If $h - 1 \geq 0$ and $\phi(H(-1)) > \phi(G)$ then set $\delta := -1$
   2.4 While $0 \geq h + \delta \geq d_i$ and $\phi(H(\delta)) > \phi(G)$ do
      2.4.1 Set $G := H(\delta)$, set $h := h + \delta$
   2.5 If $\phi(G) > f$ then go to step 1
3. Return G

In the procedure LO_2, the size of each buffer is incremented or decremented as long as this yields some improvement in the goal function. The buffers are chosen for optimization in a random order, independently for each call to local optimization procedure in the GA. Note that LO_2 terminates in a local optimum w.r.t. the neighborhood defined by a ball of radius 1.

The general scheme of the GA is as follows (in the case of GA_0 the identity mapping on genotypes substitutes the LO procedure).

Genetic algorithm with local optimization procedure LO

1. For $j := 1$ to M do:
   1.1 Generate a random genotype G
   1.1 Set $G^{(j)} := G$
2. For $t := 1$ to t_{max} do:
   2.1 Select parent genotypes $G^{(q)}$ and $G^{(r)}$ using a selection operator
   2.2 Produce the offspring G' and G'' from $G^{(q)}$ and $G^{(r)}$ using crossover operator
   2.3 Compute $G^* := LO(Mut(G'))$
   2.4 If G^* is already present in P, set $G^* := Mut(G^*)$
   2.5 Choose $1 \geq k \geq M$ such that $\phi(G^{(k)}) = \min_{1 \geq j \geq M} \phi(G^{(j)})$ and if $\phi(G^{(k)}) \leq \phi(G^*)$ then set $G^{(k)} := G^*$.

2.3 Compute $G^{**} := LO(Mut(G'))$
2.4 If G^{**} is already present in P, set $G^{**} := Mut(G^{**})$
2.5 Choose $1 \geq k \geq M$ such that $\phi(G^{(k)}) = \min_{1 \geq j \geq M} \phi(G^{(j)})$ and if $\phi(G^{(k)}) \leq \phi(G^{**})$ then set $G^{(k)} := G^{**}$.

3. Return the best solution from P.

In step 2.1 we use the s-tournament selection operator which randomly (with uniform distribution) chooses s individuals from current population and selects the best one of them as a parent. The crossover operator in step 2.2 is implemented according to the standard one-point crossover scheme, where a random position χ is chosen uniformly ranging from 1 to M, and the resulting genotypes are formed from $G^{(q)}$ and $G^{(r)}$ as follows:

$$G' = (G_1^{(q)}, \ldots, G_{\chi-1}^{(q)}, G_\chi^{(r)}, \ldots, G_M^{(r)}), G'' = (G_1^{(r)}, \ldots, G_{\chi-1}^{(r)}, G_\chi^{(q)}, \ldots, G_M^{(r)}).$$

This operator is applied with a fixed probability P_{cross}, otherwise the parent genotypes are copied to G' and G'' without any changes. The crossover is able to combine substrings form parent genotypes, so that with some probability the substrings responsible for efficient functioning of a production line in parent solutions may be inherited together in an offspring.

Given a genotype G, the mutation operator produces a genotype $Mut(G) = (G_1+\xi_1, \ldots, G_n+\xi_n)$, where ξ_i is an integer value, uniformly distributed from $\max(-G_i, -\delta)$ to $\min(d_i - G_i, \delta)$. Parameter δ defines the amount of random changes caused by the mutation and it is chosen experimentally.

2.4. 'Big Valley' or 'Massif Central' Local Optima Structures

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

1. Values of the objective function in the local optima tend to deteriorate with increasing distance to the global optimum (i.e. there is a correlation of objective function in the local optima with the 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 GA population 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 experimental studies, e.g. of Boese et al. (1993); Hains et al. (2011). Note that while Hains et al. (2011) do observe the 'big valley' in the TSP, they also notice multiple funnels, or clusters of local optima in the fitness-distance plot, and study how the crossover combines useful traits of solutions from different clusters. Another capacity of crossover is demonstrated by the theoretical analysis in the case of the Jump benchmark (see e.g. Dang et al. (2016)), and may be useful, when the global optimum is relatively far from the set of high-quality local optima (i.e. the second condition of the 'big valley' is not quite relevant). To sum up, identification of the 'big valley' / 'massif central' structure, or the absence of such structure, is a good starting point for analysis of GA performance on a fitness landscape.

In this paper, we are mostly interested with the instances where the 'big valley' / 'massif central' is hardly applicable, in particular, the instances where we see several clusters of high quality local optima or the best known solution is relatively far from the nearest cluster of local optima. We study how the clustered fitness-distance diagram corresponds to performance of crossover-based GAs. Besides that, for some instances we describe the features of production line, which correspond to different clusters of the local optima.

3. Computational Experiments

3.1. Problem Instances Used in Computational Experiment

For computational experiments we use the following series of problems:

- The AS series consists of instances as.1-as.8 based on the lines 1,2,6,7,8 Ancelin and Semery (1987) with real data from Renault production. A distinctive feature of lines 7 and 8 is the presence parallel sections (see Figures 2, 3).

Table 1: Basic parameters of the series

Series	Number of Lines	Number of Machines	Buffer sizes
AS, AS'	5	4 – 14	0-2000
BN5, BN5'	10	5	10
VP, VP'	4	5	10
GSD', GSP'	3	7–12	50–500

- Instances bn5.1-bn5.10 of series BN consist of 10 lines from Dolgui et al. (2002) with $m = 5$. There are two *bottlenecks* in each line. Here by bottleneck we mean a line section of two relatively slow machines and a buffer between them.

- Problems vp6.9-vp6.10, vp7.9-vp7.10 of series VP are defined on serial five-machine lines of Vouros and Papadopoulos (1998). These problems are of knapsack type, where the amount of buffer space is limited from above.

- Series AS', BN' and VP' are identical to AS, BN and VP, except that the inventory cost $Q(H)$ is set to zero.

- Series GSD', GSP' are defined on the basis of seven, ten and twelve-machine lines described in Gershwin and Schor (2000). Problems of series GSD' are of the knapsack type, where the amount of buffer space is limited from above by the value from Gershwin and Schor (2000) (they are called problems of dual type there). The optimization criterion for problems in GSP' series is formulated similarly to instances of series AS (they correspond to the primal type problems in Gershwin and Schor (2000)).

Basic parameters of these series are summarized in Table 1. More details about the optimization criteria, parameters of machines and buffer limits may be found in Section 1 of Supplementary Materials.

3.2. Testing the 'Massif Central'/'Big Valley' Conjecture

In what follows, the distance to a global optimum or to the best known solution will be calculated on the basis of the norm ℓ_1, i.e. a distance between any two vectors x, y is $||x - y||_{\ell_1} = \sum_{i=1}^{n} |x_i - y_i|$. To verify the "massif

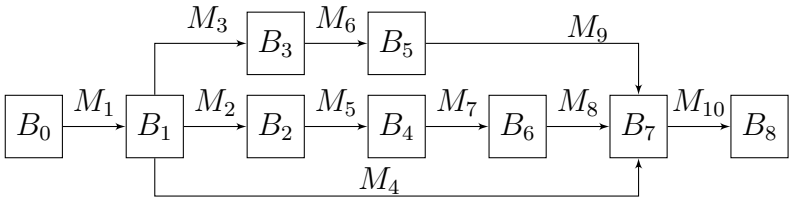

Figure 2: Structure of line in instance as.7

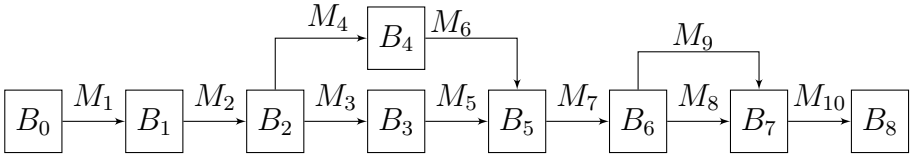

Figure 3: Structure of line in instance as.8

central" structure, we developed a method for counting the number of integer points in the intersection of a given ball of norm ℓ_1 with the parallelepiped of the search space

$$X = \{x \in Z_+^n \mid 0 \leq x_i \leq d_i,\ i = 1, \ldots, n\}$$

of the buffer allocation problem whose faces are parallel to the coordinate planes. The method is based on reduction of this counting problem to a combinatorial formulation, using the generating functions as suggested by Sachkov (1982). More details may be found in the appendix.

Series of experiments were carried out on the above mentioned instances for verification of the 'massif central'/'big valley' conjecture, when the objective function is defined by formula (1). To find a local optimum, we used the local search algorithm denoted LSA. At each iteration, the LSA searches through the neighborhood of radius 1 in norm ℓ_1 around the current solution. If an improving feasible solution in terms of the objective function is found in the neighborhood, then it becomes the new current solution. The process continues as long as an improvement can be found. (This is the termination condition of the LSA.) Starting from any feasible solution, the LSA moves iteratively to a local optimum, i.e. a solution that does not have an improving neighbour.

In each run, the local search algorithm starts at a randomly generated solution H whose elements h_i are chosen with the uniform distribution between 1 and d_i. This procedure was repeated 1000 times as in Boese et al.

Table 2: Evaluation of the ball encompassing local optima in series AS and AS'

		Series AS	Series AS'
instance	$\|X\|$	$\frac{\|\Omega\|}{\|X\|}$	$\frac{\|\Omega\|}{\|X\|}$
1	5.74E+05	0.51	0.66
2	9.47E+11	0.33	0.32
6	2.89E+22	0.51	0.68
7	9.75E+07	0.67	0.95
8	5.23E+08	0.00	0.00

Table 3: Evaluation of the ball encompassing local optima in series VP and VP'

		VP	VP'
instance	$\|X\|$	$\frac{\|\Omega\|}{\|X\|}$	$\frac{\|\Omega\|}{\|X\|}$
6.9	1.00E+04	0.61	0.61
6.10	1.46E+04	0.66	0.66
7.9	1.00E+04	0.74	0.74
7.10	1.46E+04	0.68	0.66

(1993). Based on this sample, we calculated the total number of admissible solutions $|\Omega|$ in the minimal ball encompassing all the local optima found. In our case, the ball was chosen in the norm ℓ_1, centred in the best found local optimum.

Tables 2-4 contain the cardinality of the entire search space X, and the fraction of cardinality of Ω, computed as described above, to the cardinality of X. As can be seen from the tables, the second part of the 'massif central'/'big valley' conjecture is not valid in this case. Note that in problem as.8, only one local optimum was found in all runs of the local search. Therefore we exclude this problem from the subsequent analysis.

The first part of the 'massif central'/'big valley' conjecture is about the correlation $\rho(\varphi(H'), r(H', H^*))$ of the value of objective function at local optima $\varphi(H')$ to the distance $r(H', H^*)$ to a global optimum H^*. Our experiments suggest that there is a negative correlation ρ, and all values of the correlation are significantly different from 0. These correlation coefficients are listed in Tables 5 and 6. The confidence intervals are computed with a confidence level of 95% using Fisher's z-transformation. The instance as.3 is

Table 4: Evaluation of the ball encompassing local optima in series BN and BN'

		Series BN	Series BN'
instance	$\|X\|$	$\frac{\|\Omega\|}{\|X\|}$	$\frac{\|\Omega\|}{\|X\|}$
1	1.94E+05	0.15	0.24
2	1.94E+05	0.03	0.00
3	1.94E+05	0.43	0.24
4	1.94E+05	0.10	0.00
5	1.94E+05	0.35	0.00
6	1.94E+05	0.35	0.24
7	1.94E+05	0.12	0.20
8	1.94E+05	0.01	0.26
9	1.94E+05	0.32	0.09
10	1.94E+05	0.00	0.00

not present in Table 5 because only one local optimum was found in it.

Figures 4 and 5 show the fitness-distance diagrams of local optima for problems bn5.1 and as.6 which have large ratios $\frac{|\Omega|}{|X|}$. In fitness-distance diagrams, the vertical axis shows the value of objective function of a local optimum, and the horizontal axis is the distance d (based on the ℓ_1 norm) to the best-known solution. Each point represents one or several local optima with equal distance to the best-known solution and equal fitness value. These figures demonstrate two ways the local optima may occupy a large fraction of the search space in our experiments: (i) either with a non-uniform structure which splits into several small clusters in the neighborhood of the best known solution (e.g. as.6), or (ii) with the whole structure grouped into several clusters and several separate local optima (e.g. bn5.1, bn5.3).

The buffer sizes in the best found local optimum and in a cluster of near-optimal solutions for problem bn5.1 (this cluster is indicated by a box in Figure 4) are presented in Table 7. The table shows that while the best incumbent has most of the buffer space in the first and the last position (next to the bottleneck machines), the solutions of the near-optimal cluster have large buffers before the last position. A similar divergence in solutions structure may be observed for the instance bn5.3 in Table 6 in Section 2 of Supplementary Materials. Our attempt to identify a structural difference between the two clusters presented in Figure 6 did not lead to any conclusion (all maximal buffer sizes were similar and all minimal buffer sizes were close

Table 5: Fitness-distance correlation for local optima in series AS, AS', VP and VP'. Column ρ contains the values of $\rho(r(H', H^*), \varphi(H'))$. Column "conf. int." contains the 95% confidence intervals.

instance	Series AS		Series AS'	
	ρ	conf. int.	ρ	conf. int.
1	-0.8710	(-0.885,-0.855)	-0.9053	(-0.916,-0.893)
2	-0.5085	(-0.553,-0.461)	-0.3590	(-0.412,-0.304)
6	-0.5943	(-0.633,-0.553)	-0.6081	(-0.646,-0.567)
7	-0.9306	(-0.938,-0.922)	-0.9001	(-0.911,-0.888)
instance	Series VP		Series VP'	
	ρ	conf. int.	ρ	conf. int.
6.9	-0.6706	(-0.703,-0.635)	-0.6862	(-0.718,-0.652)
6.10	-0.7601	(-0.785,-0.733)	-0.7695	(-0.794,-0.743)
7.9	-0.8732	(-0.887,-0.858)	-0.8871	(-0.899,-0.873)
7.10	-0.8897	(-0.901,-0.876)	-0.8599	(-0.875,-0.843)

Table 6: Fitness-distance correlation for local optima in series BN and BN'. Column ρ contains the values of $\rho(r(H', H^*), \varphi(H'))$. Column "conf. int." contains the 95% confidence intervals.

instance	Series BN		Series BN'	
	ρ	conf. int.	ρ	conf. int.
1	-0.8562	(-0.872,-0.839)	-0.9259	(-0.934,-0.917)
2	-0.9999	(-0.999,-0.999)	—	—
3	-0.8233	(-0.842,-0.802)	-0.9745	(-0.977,-0.971)
4	-0.9999	(-0.999,-0.999)	—	—
5	-0.9765	(-0.979,-0.973)	-0.9999	(-0.999,-0.999)
6	-0.8499	(-0.866,-0.832)	-1.0000	(-1.000,-1.000)
7	-0.9959	(-0.996,-0.995)	-0.9802	(-0.982,-0.978)
8	-1.0000	(-1.000,-1.000)	-0.8576	(-0.873,-0.840)
9	-0.8529	(-0.870,-0.835)	-0.9984	(-0.999,-0.998)

Table 7: Buffer sizes in local optima of bn5.1

best incumbent	selected cluster	
buffers value		
	min	max
19	15	18
0	0	0
3	16	19
20	20	20
function value		
21390.379	21283.109	21320.953

to zero).

The possible causes for emergence of clusters will be considered in more detail in Subsection 3.4. Note that other problems of series BN5 and BN5' contained at most two local optima, except for bn5.3 where 9 local optima were found. Instances of series VP and VP' contained hundreds of local optima, structured as a single uniform cluster.

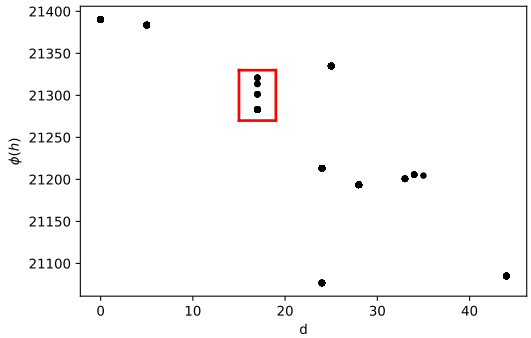

Figure 4: The set of local optima obtained in bn5.1. Results of 1000 runs of the local search.

3.3. Aggregation vs Simulation

In order to estimate the precision of aggregation approach in evaluation of the local optima quality, as an alternative to the aggregation, we consider the simulation method based on Petri nets (Dolgui and Svirin, 1995). Although this method is more time consuming, in principle it allows to approximate the

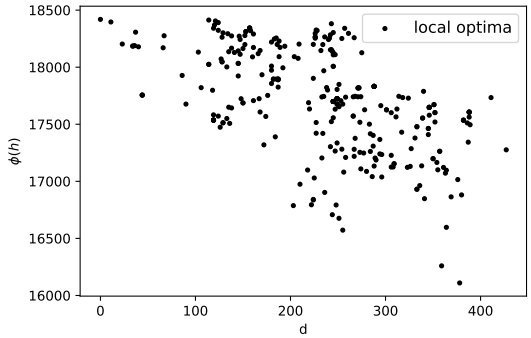

Figure 5: The set of local optima obtained in as.6. Results of 1000 runs of the local search.

true production rate with any required precision, which follows from the existence of stationary distribution and the law of large numbers (Dolgui et al., 2017). Our supplementary analysis on the set of local optima obtained in all experiments with series AS shows that the throughput computed using the simulation (with 10^5 independent simulation runs) deviated from the aggregation result, on average by less than 6%. However, due to the plateaus of the function $R(V)$ (see Section 1 of Supplementary Materials), the effect of these errors on the objective evaluations may be less than 6%. This is illustrated in the fitness-distance plot on Figure 6. Here the black dots represent the local optima obtained in 300 independent runs of the local search on as'.7, using aggregation approach (Dolgui, 1993) for fitness evaluation. The grey dots represent the same set of solutions but with the objective function calculated by means of simulation (averaged over 10^5 independent simulation runs). It can be seen that the fitness based on the simulation is close to the fitness based on the aggregation in this case.

3.4. Local Optima Clusters

It can be seen from the fitness-distance plots that in some problems the entire set of local optima splits into clusters. Note that just because two points are almost equally distant from the best-known solution does not mean they are close to each other. In what follows, we do not mean clusters in terms of Euclidean or some other distance on the search space, but in terms of images of the local optima on the fitness-distance plot. Let us consider the reasons which may cause such clustering.

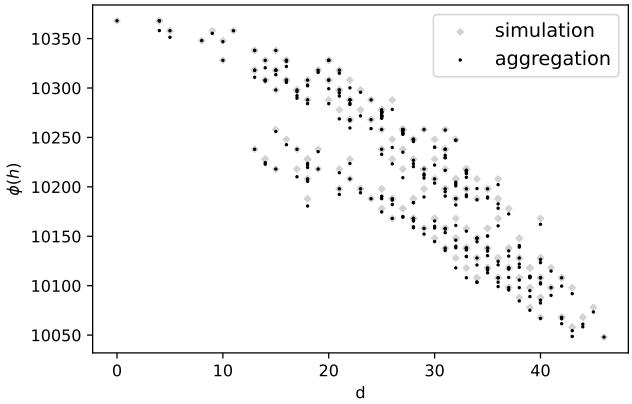

Figure 6: Local optima obtained in as'.7 using aggregation approach and the same solutions re-evaluated with simulation. The black dots correspond to the aggregation approach, the gray dots are from simulation.

3.4.1. Two-Machine Line Symmetry

One of the reasons for clustering effect is the two-machine line symmetry. This property has been established by Levin and Pasjko (1969) for a serial line, consisting of two machines and a buffer between them. If we swap the parameters of the first and the second machine in two machines and one buffer problem so that the input buffer of the line becomes its output, and the output buffer becomes the input, then the line performance will not change. Clustering of the optima in buffer allocation problems for lines with $n \geq 3$ buffers may occur as a consequence of the line symmetry effects if a line aggregation algorithms like those from Ancelin and Semery (1987); Dolgui (1993) are used to evaluate the line. Note that if such aggregation procedures are used, then for any serial line, where all parameters of machines and buffers are symmetric about the centre of the line, the results of evaluation will be identical, if the buffer sizes of any solution H are mapped symmetrically about the centre of the line. This symmetry property may lead to local optima clusters as illustrated by the following example.

Consider a three-machine serial line denoted t.1 with the 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 $R(V(H)) = 2570$. The set of global optima (see Fig. 7) of this example splits into two clusters. The first cluster contains solutions $H^1 = (1,2)$ and $H^2 = (1,3)$, and the second cluster has the solutions $H^3 = (2,1)$ and $H^4 = (3,1)$.

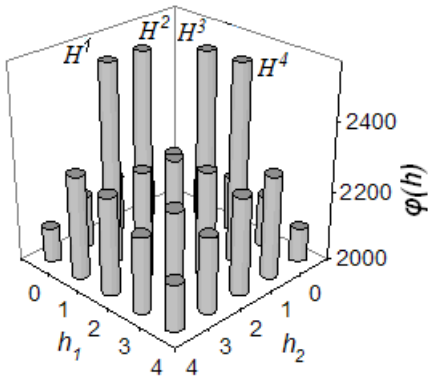

Figure 7: The objective function for problem t.1

3.4.2. Bias Due to Different Reduction Sequences

The aggregation rules from (Dolgui, 1993) and Ancelin and Semery (1987) are based on a simplifying assumption that two neighbouring machines may be substituted by a single machine, which has the same average time to failure and average time to repair as the two-machine line consisting of the pair of machines under consideration. Every time the bottleneck pair of machines is chosen for aggregation, so the ordering of machines for aggregation, depends on the buffers sizes. With different buffer allocations, the reduction rules in the aggregation method may be applied to the machines in different sequences, but different sequences have different precision in result, therefore the production rate estimation and the whole objective function may be given a bias, depending on the reduction sequence. Thus different sequences of reduction rules may create different clusters of local optima as a result of the methodological bias.

3.4.3. Parallel-Series Line Structure

Clustering effect is noticeable with parallel sections of production lines, see e.g. the plot for as.7' in Figure 6 above. This effect can be justified by the fact that there are several paths in lines with a parallel structure from start buffer to end buffer. If there are two parallel paths that are similar in their network structure (machine parameters and their connections), then in a local optimum some buffers on the first path may be given relatively large sizes, and some buffers in the second one may have relatively small size. However it is possible to obtain another local optimum of similar quality, where the buffers of the first path are given small size and the buffers of the second one are large. The two local optima would be located at a large

distance from each other in the search space, which may place them into different clusters.

3.4.4. Population Clustering in Genetic Algorithms

Solution clustering plays an important role if it occurs in populations of genetic algorithms (see e.g. Hains et al. (2011)). The goal of this subsection is to study how the local optima clustering or partial satisfaction of the 'massif central' conditions influences the performance of genetic algorithms described in Subsection 2.3. In these experiments, the GAs have the following parameters: population size $M = 50$, tournament size $s = 5$, mutation parameter $\delta = 5$, crossover probability $P_{cross} = 0.5$, which were chosen in (Dolgui et al., 2002) empirically on the basis of preliminary experiments.

Let us first consider the objective function including the inventory costs. Figures 8 and 9 show the fitness-distance plots for populations of the GA_0 after 1000 iterations for instances bn5.1 and as.6 respectively. Multiple clusters are clearly present in both cases. Problem as.6 turns out to be the hardest among the considered benchmarks, so it will be used as an illustrative example to compare different versions of the GA. Figure 10 shows the fitness-distance plots for the GA with local optimization heuristic after 1000 iterations for the instance as.6. One can see the impact of the local search usage in the GA: with this heuristic only the best solutions of the top four clusters from Fig. 9 are present in Fig. 10. When the computation of this GA continued up to 10^5 iterations, the whole population concentrated in the near-optimal cluster, located within the distance at most 14 to the best-known solution. Population clustering in vicinity of local optima was also observed in the case of bn5.1 instance, see Fig. 11 (here the global optimum is not found yet, but the GA individuals start exploring the cluster of near-optimal solutions). A similar clustering was also observed in the case of the objective function with zero inventory costs, see Fig. 12.

3.5. Improving the Performance of Clustered Populations

After falling into one of the clusters, a sequence of points generated by a metaheuristic algorithm, based on the local search principles (simulated annealing, tabu search, evolutionary algorithms etc.) usually remains within the cluster until the end of the run. This type of behaviour may be mitigated by the random restart procedures, sensitive to stagnation of the search process (Eremeev, 2019; Hampson and Kibler, 1993), by more intense usage of crossover and/or mutation (Dang et al., 2016; Doerr and Rajabi, 2023; Hains

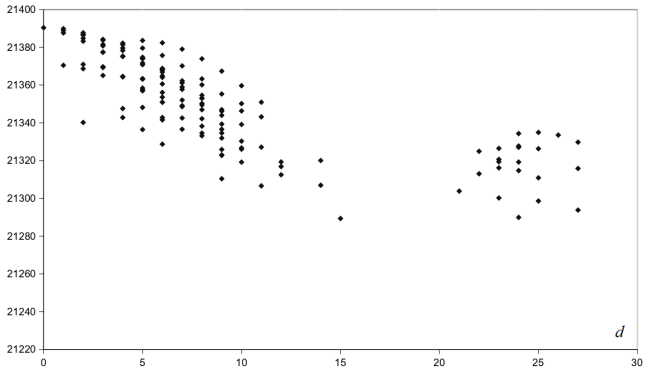

Figure 8: Population of the GA_0 at $t = 1000$ on instance bn5.1

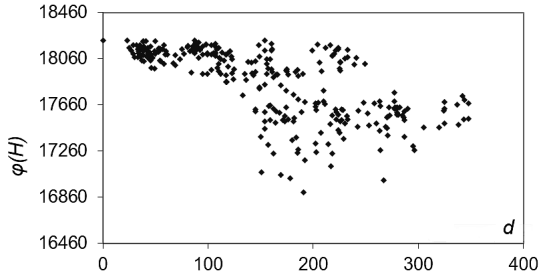

Figure 9: Population of the GA_0 at $t = 1000$ on instance as.6

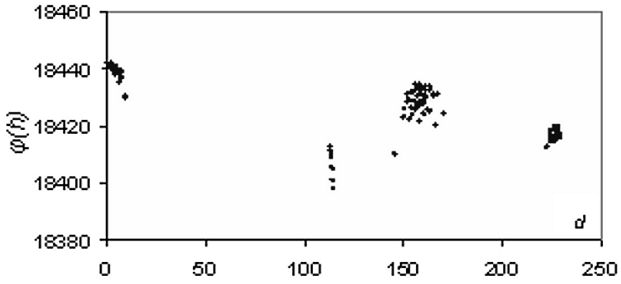

Figure 10: Population of the GA_2 at $t = 1000$ on instance as.6 (note that here the scale is different from that in Figures 5 and 9)

et al., 2011; Rajabi and Witt, 2020) or by appropriate selection mechanism in the case of non-elitist evolutionary algorithms (Dang et al., 2021; Lehre and Qin, 2022).

Here we evaluate whether the crossover is helpful to overcome the gaps between the clusters of local optima in the case of buffer allocation problem (the global optimum being considered as a cluster of size 1). Although a complete analysis of causes for efficiency of crossover is beyond the scope

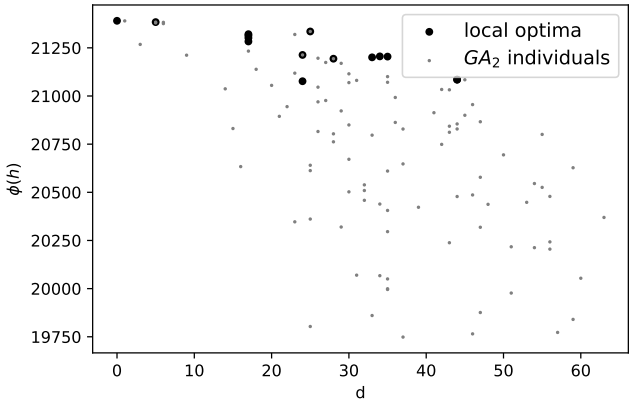

Figure 11: Population of the GA_2 at $t = 1000$ on instance bn5.1 and the set of local optima

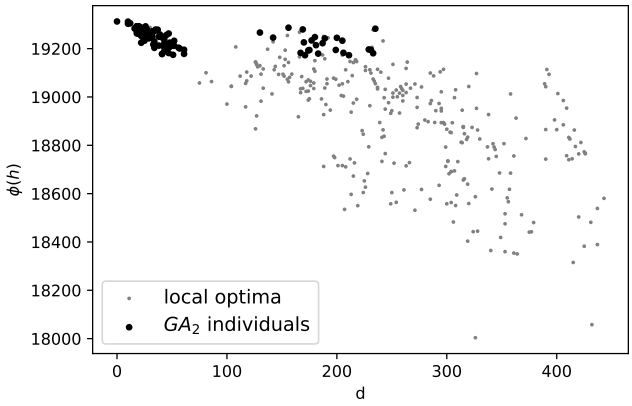

Figure 12: Population of the GA_2 at iteration $t = 1000$ on instance as.6' and the set of local optima, the case of zero inventory costs.

of this paper, we have compared three settings of crossover probability for GA_1 and GA_2, namely $P_{\text{cross}} = 0$, 0.5 and 1 on instances as.2, as.6, and as.7. Figures 13 and 14 show the average solution quality on the output of GAs after $3 \cdot 10^6$ fitness evaluations without the inventory cost and with it.

It can be seen that for the instances as.6 and as.6', the setting $P_{\text{cross}} = 1$ gives the best results. Note that as.6 and as.6' have the most non-uniform fitness-distance diagrams in series AS and AS'. In the case of as.6 and as.6', the difference between the fitness on the output of GA_2 with $P_{\text{cross}} = 0$ and $P_{\text{cross}} = 1$ is statistically significant with level $p < 0.05$ (shown by the U-test), while for the instances as.1, as.2, as.7, as.1', as.2', and as.7', where

Table 8: results of experiments with and without crossover for GA_1 for series GSP' and GSD'

instance	obtained average value		U-test
	$P_{\text{cross}} = 0$	$P_{\text{cross}} = 1$	p-value
gsd'_7	8183.572	**8184.951**	0.5533
gsd'_10	555810.830	**555853.039**	0.3870
gsd'_12	622172.286	**622260.543**	0.1768
gsp'_7	6999917.673	**6999917.810**	1.33E-04
gsp'_10	**6999554.071**	6999552.627	8.56E-03
gsp'_12	7006138.988	**7006453.218**	8.10E-40

the local optima were shaped as a uniform structure, and the GA population clustering was not observed, the usage of crossover did not make a statistically significant difference.

On the series BN, the instance bn5.1, which has the largest ratio $\frac{|\Omega|}{|X|}$ (see Table 4) and whose local optima structure splits into several small clusters (see Figure 4), was the only case where the usage of was associated with statistically significant difference in U-test ($p < 0.05$).

Similar results have been observed on series GSP' and GSD', see Tables 8 and 9. These tables present the average fitness obtained in 300 runs of the GAs with the same settings as in the experiments above. Again, as seen from Table 9, the crossover improves the average fitness obtained, and difference between the fitness on the output of GA_2 with $P_{\text{cross}} = 0$ and $P_{\text{cross}} = 1$ is statistically significant with level $p < 0.05$ on all instances, except for gsd'_7. Note that most of the fitness-distance diagrams of these instances either show a large gap between the best-found local optimum and the nearest cluster, or show several clusters of high-quality local optima with negative fitness-distance correlation, with the exception for the diagram of gsd'_7 (see Figures 9-14 of the supplementary materials). The results of the GA_1 in Table 8 are not so conclusive, probably due to a different interaction of the crossover with another local search procedure.

We conjecture that this behaviour is based on the same role of crossover as discussed in (Dang et al., 2016) where it was demonstrated that crossover can improve the GA performance, combining useful traits of distant local optima from one cluster and thus producing a globally optimal offspring at a significant distance from the cluster where the population is concentrated.

Table 9: results of experiments with and without crossover for GA_2 for series GSP' and GSD'

instance	obtained average fitness		U-test
	$P_{\text{cross}} = 0$	$P_{\text{cross}} = 1$	p-value
gsd'_7	8179.7893	**8182.234**	0.1819
gsd'_10	556676.734	**556806.290**	4.78E-08
gsd'_12	623000.641	**623188.267**	5.94E-24
gsp'_7	6999917.813	**6999917.933**	5.64E-06
gsp'_10	6999555.995	**6999558.587**	4.77E-05
gsp'_12	7006514.089	**7006538.53**	1.59E-36

Another reason for the improved results with $P_{\text{cross}} = 1$ in our experiments could be that some instances just require greater difference between the offspring and parents, which may be obtained by a greater intensity of mutation without a crossover. In order to exclude this option, we performed additional experiments where the mutation rate was optimized specifically for bn5.1 and as.6, which resulted in settings $\delta = 3$ and $\delta = 8$ respectively. So only the case of as.6 with $\delta > 5$ required further consideration. A new experiment with $\delta = 8$ for comparison of GA_2 with $P_{\text{cross}} = 0$ and $P_{\text{cross}} = 1$ on as.6 indicated a statistically significant difference in fitness of the obtained results, and an advantage of $P_{\text{cross}} = 1$ in average fitness. These additional experiments support our conjecture about the positive effect of crossover in GA_2.

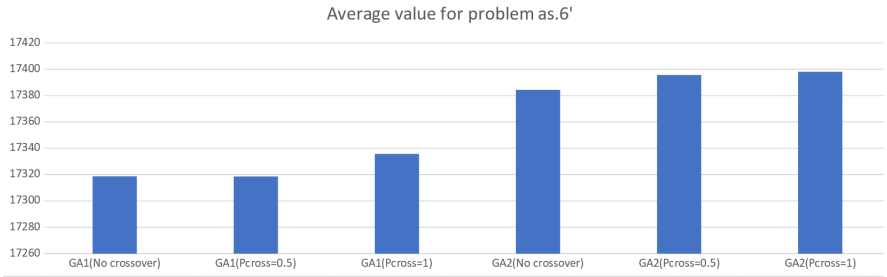

Figure 13: Average fitness after $3 \cdot 10^6$ tentative solutions on as.6' (without inventory cost).

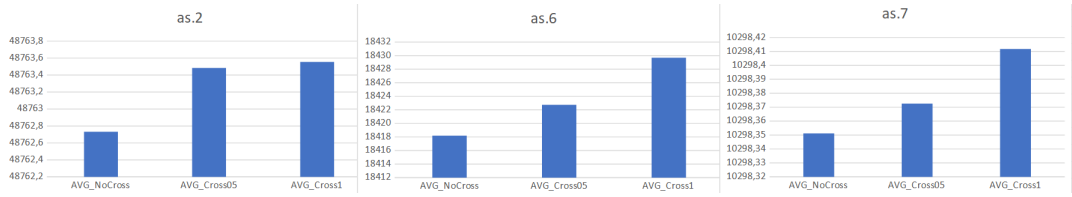

Figure 14: Average fitness after $3 \cdot 10^6$ tentative solutions with inventory cost.

4. Conclusions

The present paper extends the research on buffer allocation problems from the fitness landscape prospective, based on the production system assumptions of Coillard and Proth (1984). On the basis of computational experiments we have shown that the 'massif central' structure is present in the buffer allocation problem, but only in part: While the negative correlation between the objective function in local optima and their distance to the global solution is present, yet the concentration of all local optima in a tiny fraction of the search space is not confirmed.

The observed structures of fitness landscape appear to be similar to those in (Hains et al., 2011), and the crossover operator helps the GA_2 from (Dolgui et al., 2002) to locate the high quality local optima in hard instances of buffer allocation problem. We study how the clustered fitness-distance diagram corresponds to performance of crossover-based GAs. For some problem instances we describe the features of production line, which correspond to different clusters of the local optima.

In view of the experimental results from Section 3.5 we suggest that the genetic algorithm GA_2 is used with crossover probability $P_{\text{cross}} = 1$. We conjecture that if the knowledge of the problem symmetries or population clustering is built into the GA operators, this will simplify the transition between clusters and improve the GA performance on buffer allocation problems.

We expect that the main observations presented in this paper will hold true for a number of resource allocation problems of similar kind in other areas. In particular, the buffer allocation problems with information blocking (Buzacott and Shantikumar, 1993) which emerge in the information processing systems optimization as well as in optimization of transportation systems with unreliable communications, may have a similar fitness landscape structure as discussed in this paper, and therefore may be treated efficiently

by the similar optimization algorithms. The fitness landscape properties of buffer allocation problems for other practically valuable production line designs and buffer usage policies, such as production systems with rework loops, merging and splitting, require further investigation.

Data Availability Statement

The data and software that support the findings of this study are openly available at github:
https://github.com/DES2023/GeneticAlgorithm.git
https://github.com/DES2023/LocalSearch.git
https://github.com/DES2023/Problems.git

Acknowledgements

The authors are grateful to Dr Jean-Marie Proth for attracting their attention to buffer allocation problem and for his helpful comments at the initial stage of this research.

The computations were performed using the server of Omsk Department of Sobolev Institute of Mathematics.

Disclosure

No potential conflict of interest was reported by the authors.

Funding

A. Dolgui was supported by the ANR (French national agency for scientific research) project ANR-21-CE10-0019 "ReconfiDurable". A.V. Eremeev was supported by Russian Science Foundation grant 21-41-09017.

Appendix A.

This appendix presents the method for counting the number of integer points in the intersection of a given ball of norm ℓ_1 with a parallelepiped whose faces are parallel to the coordinate planes, containing the centre of the given ball. The method is based on reduction of this counting problem to a combinatorial formulation, using the generating functions as suggested

by Sachkov (1982). Given a set of non-negative integers $w_1, \ldots, w_n, w'_1, \ldots, w'_n$, let us define the set Ω as follows:

$$\Omega = \{x \in Z^n : \|x\|_{l_1} \leq r, -w'_j \leq x_j \leq w_j,\ j = 1, 2 \ldots, n\}. \tag{A.1}$$

Our goal is to find the cardinality of Ω.

For any $i \geq 0$ we can consider the set Ω_i, which consists of the points with norm ℓ_1 equal to i, such that the component j of these points belongs to the interval from w_j to w'_j, i.e.

$$\Omega_i = \{x \in Z^n : \|x\|_{l_1} = i, -w'_j \leq x_j \leq w_j,\ j = 1, 2 \ldots, n\}. \tag{A.2}$$

Let us consider a polynomial of the following form:

$$P_{W'W}(s,t) = Q_{w'_1 w_1}(s,t) \cdot \ldots \cdot Q_{w'_n w_n}(s,t), \tag{A.3}$$

where $W = w_1 + w_2 + \ldots + w_n$, $W' = w'_1 + w'_2 + \ldots + w'_n$ and

$$Q_{w'w}(s,t) = s^{w'} + s^{w'-1} + \ldots + s^2 + s + 1 + t + t^2 + \ldots + t^{w-1} + t^w \tag{A.4}$$

for any $w = w_j$, $w' = w'_j$, $j = 1, \ldots, n$. Now $P_{W'W}(s,t)$ may be written as follows:

$$P_{W'W}(s,t) = \sum_{i=0}^{\max(W,W')} \sum_{j=0}^{i} b_{ij} s^{i-j} t^j, \tag{A.5}$$

where b_{ij} are the coefficients in the expansion of the polynomial $P_{W'W}(s,t)$. The following lemma allows to compute the size of set Ω_i by means of the coefficients b_{ij}.

Lemma Appendix A.1.

$$|\Omega_i| = \sum_{j=0}^{i} b_{ij}, \quad i = 0, \ldots, \max(W, W'). \tag{A.6}$$

Proof. Let us introduce the set M_{ij} of all strings $(z_1, \cdots, z_n, y_1, \cdots, y_n)$, such that

1. $\sum_{k=1}^{n} z_k = i - j$;

2. $\sum_{k=1}^{n} y_k = j$;
3. if $y_k \neq 0$, then $z_k = 0, \quad k = 1, \ldots, n$;
4. if $z_k \neq 0$, then $y_k = 0, \quad k = 1, \ldots, n$;
5. $z_k \leq n_k, \quad k = 1, \ldots, n$;
6. $y_k \leq n'_k, \quad k = 1, \ldots, n$.

Consider any coefficient b_{ij} in the expansion of the polynomial $P_{W'W}(s,t)$. All coefficients of the polynomial (A.4) are equal to 1, so b_{ij} equals to the number of terms of the form $s^{z_1} \ldots s^{z_n} t^{y_1} \ldots t^{y_n}$ with all possible combinations $z_1 \cdots z_n y_1 \cdots y_n$, that satisfy conditions (1)...(6) in the product (A.3) before bringing similar, therefore $b_{ij} = |M_{ij}|$.

Note that $|\Omega_i| = \sum\limits_{j=0}^{i} |\Omega_{ij}|$, where the set Ω_{ij} contains all points in Ω_i, such that the sum of negative coordinates equals j, i.e.

$$\Omega_{ij} = \{x \in \Omega_i : \sum_{k:x_k<0} |x_k| = j\}. \tag{A.7}$$

Clearly, it suffices to prove that $|\Omega_{ij}| = |M_{ij}|$ for all $i = 0, \ldots, \max(W, W')$, $j = 0, \ldots, i$. Consider the mapping $h : \Omega_{ij} \to M_{ij}$, which for any vector x assigns the string $h(x) = (z_1(x) \ldots z_n(x), y_1(x) \ldots y_n(x))$, such that:

$$z_k(x) = \begin{cases} 0, \text{ if } x_k < 0 \\ x_k, \text{ if } x_k \geq 0 \end{cases}, \quad y_k(x) = \begin{cases} 0, \text{ if } x_k \geq 0 \\ -x_k, \text{ if } x_k < 0 \end{cases}, \quad k = 1, \ldots, n.$$

Besides that, consider a mapping $g = (g_1, \ldots, g_n) : M_{ij} \to \Omega_{ij}$ such that:

$$g_k(z_1 \ldots z_n, y_1 \ldots y_n) = \begin{cases} z_k, \text{ if } z_k \geq 0 \\ -y_k, \text{ if } y_k > 0 \end{cases}, \; k = 1, \ldots, n.$$

It is easy to see that both functions are injections. □

Proposition Appendix A.2.

$$|\Omega| = \sum_{i=0}^{r} \sum_{j=0}^{i} b_{ij}. \tag{A.8}$$

Proof. Note that the sets Ω_i do not intersect by the definition. and $\Omega = \bigcup\limits_{i=0}^{r} \Omega_i$, so by Lemma Appendix A.1 we conclude that $|\Omega| = \sum\limits_{i=0}^{r} |\Omega_i| = \sum\limits_{i=0}^{r} \sum\limits_{j=0}^{i} b_{ij}$. □

The time complexity of multiplication of polynomials $Q_{w'_j w_j}(s,t)$, and $Q_{w'_{j+1} w_{j+1}}(s,t)$ is $\mathcal{O}(w'_1 w_1 w'_2 w_2)$ for each $j = 1, \ldots, n-1$, therefore, counting the number of integer points in the intersection of a given ball of radius r in terms of norm ℓ_1 with a parallelepiped $\{x : -w'_j \leq x_j \leq w_j,\ j = 1, 2 \ldots, n\}$ takes $\mathcal{O}(nr^4)$ time, which is practically acceptable for our problem instances.

References

Alaouchiche, Y., Ouazene, Y., Yalaoui, F., 2023. A fast and efficient analytical method for throughput evaluation of unreliable series-parallel production lines.

Alfieri, A., Matta, A., Pastore, E., 2020. The time buffer approximated buffer allocation problem: A row–column generation approach. Computers & Operations Research 115, 104835. doi:https://doi.org/10.1016/j.cor.2019.104835.

Ancelin, B., Semery, A., 1987. Calcul de la productivité d'une ligne integrée de fabrication. RAIRO-Autom. Prod. Inf. 21, 209–238.

Biller, S., Li, J., Marin, S., Meerkov, S., Zhang, L., 2010. Bottlenecks in bernoulli serial lines with rework. Automation Science and Engineering, IEEE Transactions on 7, 208 – 217. doi:10.1109/TASE.2009.2023463.

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

Buzacott, J.A., Shantikumar, J.G., 1993. Stochastic Models of Manufacturing Systems. Prentice-Hall, Upper Saddle River, NJ.

Chen, C.H., Lee, L.H., 2011. Stochastic simulation optimization. World Scientific Publishing Co, Hackensack, NJ.

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

Dallery, Y., David, R., Xie, X., 1988. An efficient algorithm for analysis of transfer lines with unreliable machines and finite buffers. lIE Transactions 20, 280–283.

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

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

Dang, D.C., Eremeev, A., Lehre, P.K., 2021. Non-elitist evolutionary algorithms excel in fitness landscapes with sparse deceptive regions and dense valleys, in: Proceedings of the Genetic and Evolutionary Computation Conference, Association for Computing Machinery, New York, NY, USA. pp. 1133–1141. doi:10.1145/3449639.3459398.

Dang, D.C., Friedrich, T., Kötzing, T., Krejca, M.S., Lehre, P.K., Oliveto, P.S., Sudholt, D., Sutton, A.M., 2016. Escaping local optima with diversity mechanisms and crossover, in: Proc. of the 2016 Genetic and Evolutionary Computation Conference (GECCO 2016), ACM. pp. 645–652.

De Koster, M., 1987. Estimation of line efficiency by aggregation. Int. J. Prod. Res. 25, 615–626.

Demir, L., Tunali, S., Eliiyi, D., 2012. An adaptive tabu search approach for buffer allocation problem in unreliable non-homogenous production lines. Computers & Operations Research 39, 1477–1486. doi:10.1016/j.cor.2011.08.019.

Demir, L., Tunali, S., Eliiyi, D., 2014. The state of the art on buffer allocation problem: A comprehensive survey. Journal of Intelligent Manufacturing 25, 371–392. doi:10.1007/s10845-012-0687-9.

Diamantidis, A., Lee, J.H., Papadopoulos, C.T., Li, J., Heavey, C., 2020. Performance evaluation of flow lines with non-identical and unreliable parallel machines and finite buffers. International Journal of Production Research 58, 3881–3904.

Dieleman, N., Berkhout, J., Heidergott, B., 2023. A neural network approach to performance analysis of tandem lines: The value of analytical knowledge. Computers & Operations Research 152, 106124. doi:https://doi.org/10.1016/j.cor.2022.106124.

Doerr, B., Rajabi, A., 2023. Stagnation detection meets fast mutation. Theoretical Computer Science 946, 113670.

Dolgui, A., 1993. Analyse des performances d'un atelier de production discontinue: méthode et logiciel. Research Report INRIA RR-1949.

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 Modelling and Algorithms 1, 89–104.

Dolgui, A., Eremeev, A., Kovalyov, M., Sigaev, V., 2018. Complexity of Bi-objective Buffer Allocation Problem in Systems with Simple Structure, in: Optimization Problems and Their Applications. Springer. volume 871 of *Communications in Computer and Information Science*, pp. 278–287.

Dolgui, A., Eremeev, A., Kovalyov, M.Y., 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, pp 155–165.

Dolgui, A., Eremeev, A., Sigaev, V., 2017. Analysis of a multicriterial optimization problem for bunker capacity in an industrial line. Automation and Remote Control 78, 1276–1289.

Dolgui, A., Eremeev, A., Sigaev, V., 2022. On local optima distribution in buffer allocation problem for production line with unreliable machines. IFAC-PapersOnLine 55, 1092–1097. doi:10.1016/j.ifacol.2022.09.535. 10th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2022.

Dolgui, A., Proth, J.M., 2006. Systèmes de production modernes. Tome 1: Conception, gestion et optimisation. Hermès Science, London.

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

Dubois, D., Forestier, J., 1982. Productivité et en-cours moyens d'un ensemble de deux machines séparées par une zône de stockage. RAIRO Automatique 16, 105–132.

Eremeev, A.V., 2019. A restarting rule based on the schnabel census for genetic algorithms, in: Battiti, R., Brunato, M., Kotsireas, I., Pardalos, P.M. (Eds.), Learning and Intelligent Optimization, Springer International Publishing, Cham. pp. 337–351.

Fu, M., 2002. Feature article: Optimization for simulation: Theory vs. practice. INFORMS J. on Computing 14, 192–215.

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., Schor, J., 2000. Efficient algorithms for buffer space allocation. Annals of Operations Research 93, 117–144.

Hains, D., Whitley, L., Howe, A., 2011. Revisiting the big valley search space structure in the TSP. Oper. Res. Soc. 62 , 305–312.

Hampson, S.E., Kibler, D.F., 1993. Large plateaus and plateau search in boolean satisfiability problems: When to give up searching and start again, in: Cliques, Coloring, and Satisfiability.

Heavey, C., Papadopoulos, H., Browne, J., 1993. The throughput rate of multistation unreliable production lines. Europ. J. Oper. Res. 68, 69â–89.

Kassoul, K., Cheikhrouhou, N., Zufferey, N., 2021. Buffer allocation design for unreliable production lines using genetic algorithm and finite perturbation analysis. Int. J. Prod. Res. doi:10.1080/00207543.2021.1909169. published online.

Kassoul, K., Cheikhrouhou, N., Zufferey, N., 2022. Buffer allocation design for unreliable production lines using genetic algorithm and finite perturbation analysis. International Journal of Production Research 60, 3001–3017. doi:10.1080/00207543.2021.1909169.

Lehre, P.K., Qin, X., 2022. Self-adaptation via multi-objectivisation: A theoretical study, in: Proceedings of the Genetic and Evolutionary Computation Conference, Association for Computing Machinery, New York, NY, USA. pp. 1417–1425. doi:10.1145/3512290.3528836.

Levin, A., Pasjko, N., 1969. Calculating the output of transfer lines. Stanki i Instrument , 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., Meerkov, S., 2009. Production Systems Engineering. Springer US.

Liberopoulos, G., 2020. Comparison of optimal buffer allocation in flow lines under installation buffer, echelon buffer, and conwip policies. Flexible Services and Manufacturing Journal 32. doi:10.1007/s10696-019-09341-y.

Neri, F., Cotta, C., Moscato, P., 2012. Handbook of memetic algorithms. Springer, Berlin, Heidelberg.

Oliveto, P.S., Paixão, T., Heredia, J.P., Sudholt, D., Trubenová, B., 2018. How to escape local optima in black box optimisation: When non-elitism outperforms elitism. Algorithmica 80, 1604–1633. doi:10.1007/s00453-017-0369-2.

Papadopoulos, C.T., Jingshan, L., O'Kelly, M.E., 2018. A classification and review of timed markov models of manufacturing systems. Computers & Industrial Engineering 128, 219–244.

Patchong, A., Lemoine, T., Kern, G., 2003. Improving car body production at psa peugeot citroen. Interfaces 33, 36–49. doi:10.1287/inte.33.1.36.12723.

Rajabi, A., Witt, C., 2020. Self-adjusting evolutionary algorithms for multimodal optimization, in: Proceedings of the 2020 Genetic and Evolutionary Computation Conference, Association for Computing Machinery, New York, NY, USA. pp. 1314–1322. doi:10.1145/3377930.3389833.

Reeves, C., 1999. Landscapes, operators and heuristic search. Ann. Oper. Res. 86, 473–490.

Reeves, C., Eremeev, A., 2004. Statistical analysis of local search landscapes. J. Oper. Res. Soc. 55, 687–693.

Rodrigues, N., Malan, K., Ochoa, G., Vanneschi, L., Silva, S., 2022. Fitness landscape analysis of convolutional neural network architectures for image classification. Information Sciences 609, 711–726. doi:10.1016/j.ins.2022.07.040.

Sachkov, V., 1982. Introduction to combinatorial methods of discrete mathematics. M.: Science.

Schiavinotto, T., Stützle, T., 2007. A review of metrics on permutations for search landscape analysis. Computers & Operations Research 34, 3143–3153. doi:10.1016/j.cor.2005.11.022.

Sevast'yanov, B., 1962. The problem of how bunker capacity influences averages idle time for an automated line of machines. Teor. Veroyat. Primen. 7, 438–447.

Shi, C., 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., 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.

Stadler, P.F., 2002. Fitness landscapes. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 183–204.

Stuart, K., Simon, L., 1987. Towards a general theory of adaptive walks on rugged landscapes. Journal of Theoretical Biology 128, 11–45.

Suri, R., 1989. Perturbation analysis: The state of the art and research issues explained via the gi/g/1 queue, in: Proceedings Of The IEEE, pp. 114–137.

Tan, B., 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., David, R., 1987. Performance d'une ligne composée de machines et de stocks intermédiaires. RAIRO-Autom. Prod. Inf. 21, 239–262.

Thomson, S.L., Daolio, F., Ochoa, G., 2017. Comparing communities of optima with funnels in combinatorial fitness landscapes, in: Proceedings of the Genetic and Evolutionary Computation Conference, Association for Computing Machinery, New York, NY, USA. pp. 377–384. doi:10.1145/3071178.3071211.

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

Weiss, S., Schwarz, J., Stolletz, R., 2018. The buffer allocation problem in production lines: Formulations, solution methods, and instances. IIE Transactions 51, 456–485. doi:10.1080/24725854.2018.1442031.

Xi, S., Smith, J.M., Chen, Q., Mao, N., Zhang, H., Yu, A., 2022. Simultaneous machine selection and buffer allocation in large unbalanced seriesparallel production lines. International Journal of Production Research 60, 2103–2125.