Int j simul model 24 (2025) 4, 659-670
Original scientific paper

COST PERFORMANCE OF FRACTAL PHYSICAL LAYOUTS USING MULTIFUNCTIONAL OPERATORS

Shih, Y. C.
Federal University of Technology of Paraná – Campus Ponta Grossa, Department of Production Engineering, Rua Doutor Washington Subtil Chueire, 330 – Jardim Carvalho, Ponta Grossa – PR, 84017-220, Brazil
E-Mail: chin@utfpr.edu.br

Abstract
Fractal-layout configurations can address the inadequacy of traditional layout configurations in meeting the requirements imposed by evolving market conditions. Despite advancements, literature related to fractal physical layouts remains limited. Specifically, operator management within fractal layouts has not been sufficiently investigated. This study investigates the application of multifunctional operators within fractal physical layouts, since the spatial arrangement of these operators critically influences final product costs and the efficiency of raw material processing operations. A Tabu search algorithm was developed aiming to optimize the placement of various operator types within each fractal cell and the results indicate that total operational costs generally decrease as the number of fractal cells within the layout increases. Furthermore, allocating highly specialized operators does not significantly minimize the punitive costs associated with the system.
(Received in September 2025, accepted in October 2025. This paper was with the author 1 week for 1 revision.)

Key Words: Fractal Layout, Performance, Multifunctional Operators, Simulation

1. INTRODUCTION

Market dynamics are becoming increasingly unstable, and competitiveness among organizations continues to intensify; Gunasekaran [1] observed that product life cycles have been progressively shortening. This scenario compels companies to constantly develop new products within shorter timeframes, balancing high quality with optimized costs. Therefore, production systems must feature high adaptability, enabling swift responses to shifts in consumption patterns through innovative approaches to production planning and control. However, frequent modifications to production resources on the shop floor can, depending on the operational context, undermine the economic viability of the system. Consequently, manufacturing environments must be sufficiently robust and flexible to accommodate both product-type and demand variations (Benjaafar and Sheikhzadeh [2]).

According to Saad and Lassila [3], traditional layout configurations often prove inadequate for meeting the new requirements imposed by evolving market conditions. Fractal layouts have emerged as a promising response to this challenge. As noted by Askin et al. [4], this configuration extends the classical cellular manufacturing layout, uniquely allowing each cell to potentially encompass all necessary machine types. This design provides the production system with considerable flexibility and responsiveness. Consequently, material handling between different cells can be eliminated, as all required operations might be executed within a single fractal cell. For effective unit-based production flow, the equipment within a virtual cell (temporary configuration for a specific product) should be arranged to minimize distances. Each product type is assigned to a specific virtual cell; upon completion of its processing batch, production resources can be reconfigured to form new virtual cells tailored for other products.

According to Baykasoglu [5], the strategic dispersion of distinct machine types across the shop floor represents a highly effective approach for forming virtual cells. Expanding on this, Benjaafar and Sheikhzadeh [2] argued that complete disaggregation may not always be necessary in practice. Even partial dispersion of identical machine types can yield significant benefits, particularly by reducing material handling and movement, although they acknowledged the potential advantages of complete disaggregation.

Since these initial proposals, numerous investigations have aimed to deepen the understanding of fractal-layout configurations. The first study addressing their conceptualization is attributed to Venkatadri et al. [6], who considered the interdependence among fractal cells. This characteristic highlighted the need to determine the optimal positioning of each cell to minimize product movement between them.

Askin et al. [4] analysed the performance of structurally identical fractal cells composed of non-dedicated machines (i.e., without forming specific virtual cells). Comparing this fractal approach with functional and distributed physical layouts, they demonstrated superior response times and significantly reduced internal material handling for the fractal layout. Furthermore, they emphasized that fractal cells, besides being multifunctional, operate autonomously without operational interdependencies, meaning that all necessary manufacturing operations for a product directed to a cell are completed within that cell, eliminating reliance on machines located in other units.

Montreuil [7] compared two distinct scenarios: one using identical fractal cells (in terms of type and quantity) and another applying different fractal cells. He concluded that system efficiency remained constant when using identical cells, but observed increased material handling between cells, potentially impacting overall production performance.

Saad and Lassila [3] analysed several criteria, including product distribution (balanced vs. unbalanced), cell type (identical vs. non-identical), product routing (autonomous vs. cooperative), and layout optimization (external cell positioning vs. internal equipment positioning). Their evaluation of seven distinct combinations revealed a negative correlation between the machine count and material handling, suggesting that internal material movement tends to decrease as machine count increases.

Shih and Filho [8] proposed a methodology based on similarity principles to determine the optimal positioning of potentially dissimilar fractal cells, aiming to optimize inter-cell product movement. This approach was motivated by the observation that, in certain situations, companies might lack all necessary machine types for a pure fractal arrangement owing to prohibitive costs. In such cases, strategic positioning becomes crucial to compensate for the capacity deficiencies of one cell using resources from another.

According to Shin et al. [9], fractal cells are adaptive. Considering that each cell has specific goals, they proposed a managerial framework to optimize responses to environmental changes. Aririguzo and Saad [10] developed a genetic algorithm (GA)-based procedure to facilitate fractal-cell formation for coping with turbulent and dynamic environments. Subsequently, Jaegler [11] presented a practical analysis of implementing fractal physical arrangements in an industrial environment, highlighting the challenges encountered and lessons learned.

Elaskari and Venkatadri [12] investigated the relationship between traditional and fractal cells. They addressed deficiencies in traditional cells, particularly when unexpected demand necessitates utilizing another cell. They proposed inserting a central fractal cell in the centre of a traditional layout to absorb such unforeseen demand, considering machine-acquisition and production costs. They argued that incorporating more central cells makes the system increasingly resemble a fractal arrangement, whereas fewer central cells bring it closer to a traditional configuration, concluding that a median configuration could achieve economic viability.

Shih [13] analysed the impact of production interruptions from different maintenance policies on fractal-layout performance. He simulated varying numbers of fractal cells and evaluated performance based on two batch management strategies: waiting for maintenance completion within the cell or transferring the batch to another cell before maintenance completion. Thus, he investigated the trade-offs between increased makespan and material movement.

Askin et al. [4] cited the difficulties in implementing distributed and fractal layouts, as it was necessary to obtain real-time information from the production process regarding the status of machines to form efficient routings. Sandkuhl and Kirikova [14] analysed the implementation of fractal organization in the public sector and in a private company, not discussing it lower managerial levels. They stated that the actual implementation of fractal organization involves goal orientation, self-organization, self-similarity, and an efficient information and communication system. Given the importance of technology use, Peron et al. [15] stated that, at that time, the technologies developed were limited to executing newly developed procedures in layout studies. They then discussed several recently developed technologies for industrial applications, such as mobile robots, 3D mapping, indoor positioning systems, infrared positioning systems, and others, to support the functionality of implemented layouts.

As we can see, operators play a crucial role in transforming raw materials, and their effective management is essential for optimal production-system performance. Although significant advances have been made in studying fractal physical layouts, the related literature remains limited. Despite the importance of operators, there was a lack of specific discussion regarding their role on the shop floor. A search in databases such as Compendex using the keyword "fractal physical layout" yielded only 31 articles, most of which were previously mentioned. The others are related to the application of fractal structures in fields such as solar-panel development, product distribution, and data-block development for semiconductors. A search using the keyword "fractal organization" returned approximately 233 records, but studies directly related to manufacturing, beyond those already discussed, were primarily limited to Shin et al. [9] and Jaegler [11]. A search in the Scopus database using the keyword "application of fractal organisation" identified mainly the work of Sandkuhl and Kirikova [15], who discussed the practical implementation of fractal layouts.

Notably, the issue of operator management has been largely overlooked (Vilela et al. [16]). Such authors analysed the impacts of operator, concerning their behaviour, on the performance of traditional assembly lines. According to Sabioni et al. [17], in mass customisation the aim is to produce a variety of products with lower total cost and lead time. They cited modular products as an alternative to achieve the desired variety and noted that when the design of modular products was previously communicated to the shop floor, the total cost could be reduced. In their approach, it was possible to improve the reconfigurable manufacturing system by changing the positions of machines. However, they omitted operators from the analysis. In contrast, our research differs from theirs in that its objective was to develop a robust layout capable of producing any type of product without changes. We investigated about the fractal layouts, concerning the relationship between the number of fractal cells and the required operator assignments. It was embedded in the analysis factors such as the number and types of machines each operator must control, operator qualifications, and the product type being manufactured. Most concerns are associated with the practical situations that operators face on a daily basis. The analysis seeks to ensure operations are completed without delays, while simultaneously reducing the total system cost, which constitutes the primary objective of this study. The computer modelling and simulation is extremely flexible and useful in these cases because of the ability to vary layout sizes, machines, operator´s position among other raised issues, and to collect results for comparison.

2. OBJECTIVE FUNCTION FORMULATION

Given the considerations above, we proposed the following objective function to minimize the *TC*, see Eq. (1):

$$TC = min\,\{(\sum_{k'=1}^{k}\sum_{p'_{k'}=1}^{p_{k'}}\sum_{j_{p'k'}=1}^{J_{p'k'}} c' \cdot (1+r)^{CL-1} \cdot (1+s)^{i-1} \cdot (1+w)^{cplx-1}) + (\sum_{k'=1}^{k}\sum_{p'k'=1}^{pk'}\left(t_{finished\ instant\ p'k'} - t_{arrival\ instant\ p'k'}\right) \cdot c_{CL})\} \quad (1)$$

Subject to the following constraints:

$$k = \min\{q_{m=1,2,..,M}\} \quad (2)$$

$$q_m \geq 1 \quad (3)$$

$$1 \leq p'k' \leq pk' \quad (4)$$

$$2 \leq j_{p'k'} \leq J_{p'k'} \quad (5)$$

$$t_{arrival\ instant\ p'k'} \geq 0 \quad (6)$$

$$t_{finished\ instant\ p'k'} > 0 \quad (7)$$

$$t_{operation\ p'k'j} > 0 \quad (8)$$

$$c' > 0 \quad (9)$$

$$0 \leq r, s, w \leq 1 \quad (10)$$

$$CL, i, cplx \geq 1 \quad (11)$$

$$c_{CL} \geq 0 \quad (12)$$

The first part of Eq. (1) was about the increment of training cost due the product class, qualification level and machine complexity, and the second part was about the punitive cost due to products conclusion delays. Constraint (2) defines the number of fractal cells required. Constraint (3) indicates that there must be at least one unit (q_m) of each machine type *m*. Constraint (4) specifies that fractal-cell k' must process at least one product type $p'k'$, up to a maximum of pk'. Constraint (5) stipulates that all considered products must have at least two operations ($j_{p'k'}$), up to a maximum of $J_{p'k'}$. Constraint (6) defines the raw material arrival time, which is equivalent to the order arrival time. Constraint (7) is the time when all raw material units are completed, thus obtaining all product units. Constraint (8) ensures that the duration of an operation to be executed for a product is non-zero. Constraint (9) represents the non-negativity of the general operator training cost. Constraint (10) corresponds to a percentage cost increase factor, ranging from 0 to 100 %. Constraint (11) indicates that there must be at least one product class, operator qualification type, and machine category. Constraint (12) defines the unit-time PC, ensuring it assumes positive values; this cost is null only if the product is delivered exactly at the order arrival time. Note that this study assumed that the raw material arrival time at cell k' corresponds to the order arrival time.

3. MATERIALS AND METHODS

The performance characteristics of the fractal physical arrangements in this study were assessed through computational simulations. The system comprised an Intel® Core™ i3-10100 CPU @ 3.6 GHz (* CPUs), with 4096 MB RAM. In this research work, we opted in choosing the basic programming software, because this type of software is extremely flexibility in changing the layout size, position of operators, machines to form virtual and fractal cells, and so on. While the others (visual simulators) were restricted in these cited features, requiring remodelling. For instance, in Othman et al. [18] it was possible to use one of the visual simulators to improve the effectiveness of the rewinding workshop because the operation sequence was fixed and the unfinished product passed through exactly in the same machines. Nowadays more sophisticated analysis is required. As cited by Zhang [19], an efficient production system may be reached when it is considered dynamic allocation of resources, that is, it is important that a simulation software permits to evaluate by itself which suitable resource production must be chosen as the production advances. Pascal was then chosen as the programming language, and the simulations were implemented using Dev-Pascal 2.0, which can be downloaded for free.

The method involved during the construction of the model consisted of several steps. In each step the verification was conducted by using Test Turing, which consists in evaluating whether the logical structure of the model is reproducing what was determined by the modeler. Concerning about the validation, it was not possible because we were not comparing with an existing real layout. However, we tried to use most part of input data retrieved from Shih [13] and compared the simulation results with some layout sizes presented in the literature.

Step 1 – Defining the quantity of machines: This study used data on the size (number of machines and types) of fractal physical arrangements presented in the literature (Saad and Lassila [3], Venkatadri et al. [6] and Shih [13]. Montreuil [7] suggested that the number of fractal cells (k) should be limited by the quantity corresponding to the least numerous machine type required within a cell, aligning with constraints (2) and (3).

Step 2 – Positioning of machine types within fractal cells: The dispersion of machines across the shop floor is a critical aspect of fractal-layout design. Randomly allocating machines to available locations does not guarantee an optimal layout; typically, this approach results in a high distribution degree, calculated using Eqs. (13) and (14), indicating suboptimal distribution. To illustrate the distribution degree, consider an example with three machine types (A, B, and C) and quantities: $q_A = 2$, $q_B = 1$, $q_C = 4$. For the first unit of Machine A, identify the nearest Machines B and C. Sum these distances ($S1$) and repeat this process for the second unit of Machine A to find its sum of nearest distances ($S2$) to B and C. The average distance for machine type A relative to other types is $\overline{d_A} = (S1 + S2)/q_A$. This calculation is repeated for Machines B and C to obtain $\overline{d_B}$ and $\overline{d_C}$. Next, the overall distribution degree (in distance units) is calculated as follows: $(\overline{d_A} + \overline{d_B} + \overline{d_C})/3$ (because there are three machine types in this case).

A GA was developed to help determine the fractal layout by minimizing this distribution degree. Montreuil [7] proposed the following formulation for distribution degree:

$$\phi = \sum_{m=1}^{M} \sum_{m'=1}^{q_m} \frac{\delta_{m'}}{M \cdot q_m} \tag{13}$$

$$\delta_{nm'} = \sum_{m'' \neq m}^{M} d^{*}_{m'm''} \tag{14}$$

The resulting layout solution positions same-type machines as far apart as possible, providing the flexibility desired for operating effectively in chaotic environments.

Step 3 – Product definition and cell assignment: A proportional relationship often exists between the quantity of a specific machine type and its usage frequency in production operations. However, given that fractal layouts are intended for turbulent environments, this study assumed that these proportional relationships are not strictly adhered to. Therefore, a random sequence of operations was generated for products, irrespective of the available quantity of each machine type, following the specifications in constraints (4) to (8). A key element in this study is calculating the value of each product. By definition, the value of product p' in cell k' is the product of its demand quantity and selling price, see Eq. (15):

$$Value_{p'k'} = NL_{p'k'} \cdot LS_{p'k'} \cdot PR_{p'k'} \tag{15}$$

This value is compared with those of other products within the same cell k'. The Pareto rule is then applied to classify products based on their relative value contribution.

Step 4 – Virtual cell definition: Eq. (16) identifies the potential machine sets for processing a specific product type pk' by summing the distances between the machines within the set. This calculation is repeated for all pk'. Eq. (17) computes the total distance ($D_{k'}$) for each potential virtual-cell configuration. This distance is compared to a large initial value D (where $D \gg 0$). If $D_{k'} < D$, the set of machines forming this virtual cell is stored, and D is updated ($D \leftarrow D_{k'}$). This iterative process is repeated for a specified number of iterations to identify the best virtual-cell configurations for all products within each fractal cell.

$$\left.\begin{matrix} If\ J_{p'k'} > 1 \text{ then } d_{p'k'} = \{\sum_{j=2}^{J_{p'k'}} \left|x'_{j_{p'k'}} - x'_{j_{p'k'}-1}\right| + \left|y'_{j_{p'k'}} - y'_{j_{p'k'}-1}\right|\} \\ If\ J_{p'k'} = 1 \text{ then } d_{p'k'} = 0 \end{matrix}\right\} \tag{16}$$

$$D_{k'} \leftarrow D_{k'} + d_{p'k'} \tag{17}$$

$$D_{k'} \geq 0, \text{ where } D_{k'} \text{ starts at zero} \tag{18}$$

$$x > 0 \tag{19}$$

$$x' \leq x \tag{20}$$

$$y' > 0 \tag{21}$$

$$y' \leq y \tag{22}$$

Constraints (18), (19), and (21) enforce non-negativity for distances and layout dimensions (x', y'). Constraints (20) and (22) ensure that the Cartesian coordinates (x', y') where machines are positioned do not exceed the predefined dimensions (x, y) of the fractal layout; i.e., machines cannot be allocated outside this area.

Step 5 – Operator number definition: The relationship between $q_{i_{k'}}$ and $NOO_{i_{k'}}$ is defined as Eqs. (23) and (24):

$$\sum_{i_{k'}=1}^{I_{k'}} NOO_{i_{k'}} = TQM_{k'} \tag{23}$$

$$1 \leq TNO_{k'} = \sum_{i_{k'}=1}^{I_{k'}} q_{i_{k'}} \leq \sum_{mk'=1}^{M_{k'}} q_{mk'} = TQM_{k'} \tag{24}$$

$$q_{mk'} \geq 1 \tag{25}$$

$$1 \leq i_{k'} \leq I_{k'} \tag{26}$$

$$0 \leq NOO_{i_{k'}} \tag{27}$$

$$1 \leq \sum_{i_{k'}=1}^{I_{k'}} NOO_{i_{k'}} \leq TQM_{k'} \tag{28}$$

$$M_{k'} > 0 \text{ and } M_{k'} = M \text{ for } \forall\, k' \tag{29}$$

$$q_{i_{k'}} \geq 0 \tag{30}$$

$$TQM = \sum_{k'=1}^{k} TQM_{k'} \tag{31}$$

Constraint (24) stipulates that each fractal-cell k' must have at least one operator, and the number of operators cannot exceed the total number of machines ($M_{k'}$) assigned to that cell. Constraint (25) requires at least one unit of each machine type m within cell k'. Constraint (26) mandates that at least one operator type i is required, up to the maximum number of specialties $I_{k'}$ available within cell k'. Constraint (27), allowing a null value, indicates that a specific operator i might not control any operation type, implying they may not be needed in cell k'. Constraint (28) allows for the possibility that at least one operator of type i controls all machine types in cell k' (up to $TQM_{k'}$), whereas other operators might control only one operation (hence the value 1). Constraints (29) and (30) represent non-negativity requirements, with the equality in (29) specifying that all cells k' contain all machine types M. Finally, constraint (31) ensures that the sum of machines across all fractal cells equals TQM.

Step 6 – Operator assignment optimization: Optimization of the objective function (Eq. 1) involves altering the position of each operator. To illustrate this, a simplified example for a fractal-cell k' is presented in Table I.

Table I: Sample representation of a fractal-cell k' in vector form.

5	2	1	1	4	3	5	4	2

The numbers in this vector represent the machine types, while their sequence denotes the specific position each machine occupies, determined based on the distribution degree. For this example, assume operator quantities are q_A = 2, q_B = 2, and q_C = 1, yielding five operators. Operator A represents a general operator controlling a single machine, whereas Operator C is more qualified, capable of controlling up to three machines. Table II presents the interpretation of a specific allocation under this scenario: one Operator A controls machine type 5, and the second Operator A controls only machine type 2. Regarding Operator B, one controls two machines (both of type 1), whereas the other controls machines types 4 and 3. Finally, one Operator C controls three machines: types 5, 4, and 2.

Table II: Example allocation of operator types to specific machines corresponding to fractal cell k' in Table I.

<table>
<tr><th>A</th><th>A</th><th colspan="2">B</th><th colspan="2">B</th><th colspan="3">C</th></tr>
<tr><td>5</td><td>2</td><td>1</td><td>1</td><td>4</td><td>3</td><td>5</td><td>4</td><td>2</td></tr>
</table>

Note that swapping the assignments between operators of the same type and qualification (such as the two type A operators in the example) is redundant. TS was selected for optimization because it can effectively manage such characteristics. The solution encoding represents the number of machines controlled by each operator type; for the allocation shown in Table II, this vector is [1 1 2 2 3]. This implicitly defines the required qualification level for each operator instance, based on the assumption that controlling one machine corresponds to type A, two machines to type B, and three machines to type C. Suppose an optimization step, such as a swap, modifies the machine-count vector to [2 1 2 3 1], or simply [2 2 1 2 2 3 3 3 1]. This altered configuration results in the new operator allocation detailed in Table III.

Table III: Example operator allocation resulting from a hypothetical optimization swap applied to the configuration shown in Table II.

<table>
<tr><th colspan="2">B</th><th>A</th><th colspan="2">B</th><th colspan="3">C</th><th>A</th></tr>
<tr><td>5</td><td>2</td><td>1</td><td>1</td><td>4</td><td>3</td><td>5</td><td>4</td><td>2</td></tr>
</table>

Step 7 – Simulation input data and parameters: This study utilized machine quantity and type data from existing literature to define three layout sizes for simulation: 30 machines (based on Saad and Lassila [3]), 45 machines (based on Venkatadri et al. [6]), and 90 machines (based on Shih [13]), as listed in Table IV.

Table IV: Machine types, quantities (replicas), and complexity (*cplx*) from constraint (11) defined in this study.

		Machine type									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Saad and Lassila [3]	**Quantity**	10	6	6	7	16					
	Complexity (*cplx*)	1	1	2	3	3					
Venkatadri et al. [6]	**Quantity**	4	4	4	4	4	4	3	3		
	Complexity (*cplx*)	1	1	1	2	2	2	3	3		
Shih [13]	**Quantity**	8	10	9	10	11	9	8	9	6	10
	Complexity (*cplx*)	1	1	1	2	2	2	2	3	3	3

The objective was to determine the optimal structure in terms of the number of fractal cells, operator organization within each cell, and their positioning to minimize the TC for each layout size. We opted to define as input data a range of numbers instead of traditional statistical distribution because the intention, when fractal layouts were proposed in the literature, was to operate in unstable conditions, capable in producing efficiently with random input data.

Parameter ranges were defined as follows: number of operations per product ($J_{p'k'}$) = 2–7, number of batches from ($NL_{p'k'}$) = 1–3, batch size ($LS_{p'k'}$) = 1–5, and operation duration ($t_{operation\ p'k'j}$) = 1–5 time units. Additionally, the demand arrival time per product type varied from five to ten time units after the previous arrival. Unit prices ($PR_{p'k'}$) for product types ranged from 1 to 100 monetary units. The ABC classification rule was applied to products allocated to each fractal cell, and classification was performed immediately after their balanced distribution. Based on calculated product values Eq. (15), products were sorted by relative contribution; those accounting for the top 80 % of cumulative probability were classified as category A; those contributing from >80 % up to 95 % were classified as B; and the remaining were classified as C. For computational implementation, these variables were mapped to CL: category A corresponds to CL = 3, B to CL = 2, and C to CL = 1. Similarly, parameters for machine complexity (*cplx*) and operator qualification (i) were defined: simpler machines and common operators were assigned base values (*cplx* = 1, i = 1), whereas more complex machines were assigned *cplx* = 3, and higher operator qualifications were represented by values up to i = 5. The parameters i and CL are included in constraint (11), which requires them to be ≥ 1.

It was assumed that operations for a product do not repeat, and machines operate continuously without interruptions (for maintenance or tool changes). The simulation involved 30 distinct product types, distributed equally among the fractal cells (e.g., if k = 3, then $p_{k'}$ = 10; the results are presented in Section 5).

For the GA used in layout optimization, the following parameters were employed: 100 generations, population size 40, crossover rate 80 %, and mutation rate 2 %. For the TS used in operator assignment optimization, the Tabu list size was five (pairs of swaps), and IT = 20 iterations were performed. These heuristic parameters were selected based on preliminary tests indicating reasonable convergence for the respective objective functions (distribution degree for GA, TC for TS). The cost of a common operator was c' = 1 monetary unit, constraint (9), and the unit-time PC was c_{CL} = 0.0001 monetary units per time unit, constraint (12), regardless of product type. The percentage increase factors, constraint (10), were set to $r = s = w$ = 20 %.

4. RESULTS AND DISCUSSIONS

A total of 30 instances were defined for the simulation runs, where each instance represents a specific initial configuration of operator types and their positions within a fractal cell (referred to as Pre-Tabu configurations). For example, consider Instance 1 detailed in Table V. The initial operator assignment vector [33312222221133311] required a training cost (*TRC*) of 17.19 monetary units and *PC* of 0.16 monetary units to produce the 15 product types assigned to that fractal cell, yielding a *TC* of 17.35 monetary units. Applying the TS optimization yielded an improved assignment vector [33313133222121212]. This reduced the training cost to 16.12 monetary units, while the *PC* remained unchanged, resulting in a *TC* of 16.28 monetary units.

Among the 30 initial instances (Pre-TS), Instance 10 provided the lowest *TC* (16.30 monetary units), with the operator-positioning vector [22122111113332222]. However, after applying the TS optimization across all instances, the best overall result (lowest *TC*) was achieved by Instance 9, with a *TC* of 15.86 monetary units and an operator-positioning vector of [22121133212212231]. Therefore, after TS optimization, the *TC* for the overall fractal physical arrangement was 33.61 monetary units (15.69 + 0.17 from Instance 9 and 17.30 + 0.45 from Instance 30).

Table V: Simulation results (TC) pre- and post-TS for 30 instances using data from Saad and Lassila [3] (two fractal cells, 5 × 9 layout, up to three operator types). * = best result found.

	Two fractal cells, three operator types							
	Fractal cell 1				**Fractal cell 2**			
	Pre-Tabu		**Post-Tabu**		**Pre-Tabu**		**Post-Tabu**	
Instances	***TRC***	***PC***	***TRC***	***PC***	***TRC***	***PC***	***TRC***	***PC***
1	17.19	0.16	16.12	0.16	21.12	0.39	19.35	0.45
2	18.50	0.15	17.60	0.17	19.48	0.44	18.62	0.42
3	18.47	0.15	17.30	0.17	21.70	0.37	19.26	0.43
4	18.67	0.13	17.95	0.14	20.52	0.46	18.55	0.39
5	19.35	0.12	17.43	0.14	20.15	0.40	18.75	0.43
6	19.57	0.13	19.47	0.12	20.47	0.40	19.66	0.42
7	18.19	0.14	17.35	0.16	19.78	0.39	19.66	0.37
8	17.74	0.16	16.87	0.15	19.97	0.40	18.60	0.37
9	16.55	0.17	15.69*	0.17*	20.10	0.42	18.35	0.43

10	16.13*	0.17*	15.81	0.17	19.47	0.39	17.91	0.44
11	18.62	0.15	17.80	0.15	19.79	0.47	17.77	0.41
12	18.65	0.14	17.27	0.17	21.09	0.38	18.88	0.36
13	20.46	0.13	20.06	0.13	18.47	0.46	17.42	0.46
14	19.19	0.12	18.79	0.12	20.99	0.39	18.96	0.40
15	16.70	0.15	16.14	0.17	20.76	0.40	19.15	0.45
16	19.66	0.13	17.20	0.16	19.78	0.39	18.45	0.40
17	18.88	0.14	16.07	0.17	19.98	0.38	18.64	0.40
18	16.80	0.15	16.25	0.16	20.92	0.40	18.71	0.41
19	18.82	0.13	17.54	0.16	19.62	0.39	18.44	0.42
20	19.85	0.12	19.19	0.12	21.54	0.41	20.37	0.40
21	19.42	0.14	18.32	0.16	20.20	0.36	18.87	0.40
22	20.30	0.13	19.57	0.13	19.73	0.46	18.11	0.43
23	18.07	0.14	16.76	0.15	21.08	0.36	20.36	0.39
24	17.12	0.16	16.47	0.17	18.93	0.44	18.03	0.44
25	17.81	0.15	16.54	0.15	20.70	0.39	18.51	0.40
26	17.83	0.14	16.14	0.17	21.33	0.36	19.26	0.41
27	16.49	0.17	16.34	0.17	21.15	0.35	19.44	0.39
28	18.24	0.14	16.87	0.13	18.48	0.44	17.98	0.44
29	17.81	0.15	17.03	0.13	19.74	0.41	17.74	0.44
30	18.31	0.14	18.10	0.15	18.35*	0.48*	17.30*	0.45*

Fig. 1 illustrates the final simulation results obtained using the data from Saad and Lassila [3].

```
***********Final layout****************        **********Sequence allocation of cells*********
1 1 1 5 3 4 1 3 5                              1 1 1 1 1 1 1 2 2
4 2 3 4 2 5 5 1 4                              1 1 1 1 1 1 1 2 2
5 5 5 5 2 1 5 2 1                              1 1 1 2 2 2 2 2 2
5 3 4 4 3 4 3 1 5                              2 2 2 2 2 2 2 2 2
5 1 2 5 1 2 5 5 5                              2 2 2 2 2 2 2 2 2

***********Position of operators that resulted in the lowest total cost *************
Operators in cell:1
2 2 1 2 1 1 3 0 0
3 2 1 2 2 1 2 0 0
2 3 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0

Operators in cell:2
0 0 0 0 0 0 0 1 3
0 0 0 0 0 0 0 2 2
0 0 0 2 1 2 2 1 2
2 1 2 1 2 1 3 3 2
2 3 3 1 1 2 3 2 2

**********Final total cost of fractal layout ***************
33.61
```

Figure 1: Visualization of optimized simulation results for the two-fractal-cell configuration and allowing up to three operator types.

The "final layout" panel depicts the machine arrangement obtained using the GA, which achieved the lowest distribution degree. The numbers indicate machine types (1–5). The "allocation sequence" panel shows the order of fractal-cell allocation determined using a Peano switch-back type curve (Peano-R), where numbers 1 and 2 represent fractal cells 1 and 2, respectively.

The "operators in cell 1" panel visualizes the operator types assigned within cell 1, where type 1 controls one machine, type 2 controls two machines, etc. Note that these operators are only assigned to fractal cell 1; the corresponding visualization for cell shows zero assignments, indicating that it is occupied by another group of operators designated specifically to it.

The detailed operator allocation derived from the optimized two-cell simulation (Fig. 1) for fractal cell 1 is as follows: six type-1 operators control individual machines ([1], [3], [4], [3], [5], and [5]); four type-2 operators collectively control machine pairs ([1 1], [5 4], [2 5], [2 5]), and the single type-3 operator controls a set of three machines ([1 4 5]). For fractal cell 2, eight type-1 operators control individual machines ([3], [2], [2], [3], [4], [4], [5], [1]), seven type-2 operators control the pairs ([1 4], [5 1], [5 1], [5 4], [3 5], [5 1], [5 5]), and two type-3 operators control sets of three machines ([5 3 1], [1 2 5]).

Simulations were also conducted for a three-fractal-cell configuration, while keeping all other simulation parameters identical to those used in the previous runs (Fig. 2).

```
***********Final layout****************   **********Sequence allocation of cells*********
5 3 2 4 3 4 5 5 1                          2 2 2 2 2 2 2 3 3
5 1 5 5 1 2 1 3 2                          1 2 2 2 2 2 2 3 3
1 4 3 2 1 5 5 1 4                          1 1 1 1 1 1 1 3 3
5 1 5 4 3 2 1 5 5                          3 3 1 1 1 1 1 3 3
5 4 3 2 5 5 4 1 5                          3 3 3 3 3 3 3 3 3
```

Figure 2: Visualization of optimized simulation results for the three-fractal-cell configuration and allowing up to three operator types.

In the simulation with three fractal cells, the operator-positioning vectors determined by TS were [1211221222111] for cell 1, [2123223111131] for cell 2, and [1112212333122121221] for cell 3. Specifically, for cell 1, type-1 operators controlled individual machines [5], [4], [3], [5], [3], [2], and [1]; type-2 operators controlled pairs [1 2], [1 5], and [5 4]; and one type-3 operator controlled the set [4 5 2]. In cell 2, type-1 operators controlled machines [3], [1], [5], [5], [1], and [1]; type-2 operators controlled pairs [5 2] and [3 4]; and type-3 operator controlled the set [4 5 2]. In cell 3, type-1 operators controlled machines [5], [1], [3], [4], [5], [2], [5], and [5], whereas type-2 operators controlled the pairs [2 1], [5 4], [3 5], and [4 1]. This three-cell simulation resulted in a *TC* of 32.47 monetary units, representing a decrease than the optimized two-cell cost of 33.61. These results support the observation that both training and *PC*s tend to be lower when more fractal cells are used.

Further simulations explored the impact of operator qualification levels. Simulating the two-cell layout with up to five operator types resulted in a *TC* of 35.20 monetary units. This indicates that training operators to handle more machine types increases the training cost, thereby raising the *TC*. Although employing more qualified operators capable of controlling more machines and performing operations faster could theoretically improve lead time, this did not translate into a reduction in the *PC* in these simulations. Simulating three cells with up to five operator types resulted in a *TC* of 35.70 monetary units. Although the individual performance of each cell might be better compared to the two-cell/five-type scenario, the overall *TC* increased compared to the three-cell/three-type configuration (35.70 vs 32.47). These results are synthesized in Table VI.

Simulations were performed using the input data presented by Venkatadri et al. [6] (representing a 45-machine layout in a 5 × 6 configuration); these results were presented in Table VI. They further confirm the previous observation that employing fewer types of multifunctional operators in conjunction with more fractal cells reduces the *TC* (decreasing, for instance, from 35.21 to 31.74). Consistent with earlier findings, introducing operators capable of controlling more machines increased the training cost without improving lead times. These results also indicate that using more machines (30–45) tends to lower the *TC* when the arrangement employs exactly two fractal cells. However, it was also observed that under certain configurations – specifically, when utilizing more fractal cells along with fewer types of multifunctional operators – the smaller (30-machine) physical arrangement can outperform the larger (45-machine) layout, highlighting a notable interaction effect. With larger layout (90 machines, 9 × 10 configuration) of Shih [13], the *PC* (linked to lead time) tends to be lower than that with smaller layouts, possibly owing to the increased possibilities for forming efficient virtual cells within each fractal cell. However, the two-cell configuration appears economically unviable for this large layout size. Increasing the configuration to three fractal cells significantly improves the cost performance, achieving *TC*s comparable to those found in the simulations using smaller physical arrangements.

Since we were facing with synthesized results and different performance indicators, it was not possible to conduct a specific statistical comparison. We opted to present DRB, which means Difference in Relation to the Best, comparing in percentage each *TC* to the best *TC* (31.74*).

Table VI: Comparison of TCs and components (TRC, PC) simulation results for two- vs. three-cell configurations with varying operator types using data from Saad and Lassila [3], Venkatadri et al. [6] and Shih [13]. Costs in monetary units, DRB means Difference in Relation to the Best (* = the best).

	Saad and Lassila [3]											
	Two cells					**Three cells**						
	Cell 1		**Cell 2**			**Cell 1**		**Cell 2**		**Cell 3**		
Operators types up to	***TRC***	***PC***	***TRC***	***PC***	***TC***	***TRC***	***PC***	***TRC***	***PC***	***TRC***	***PC***	***TC***
3	15.69	0.17	17.30	0.45	33.61	11.27	0.25	10.41	0.08	10.34	0.11	32.47
DRB					**+5.89 %**							**+2.30 %**
5	16.00	0.27	18.34	0.58	35.20	13.05	0.28	11.17	0.12	10.91	0.16	35.70
DRB					**+10.90 %**							**+12.48 %**
	Venkatadri et al. [6]											
3	17.41	0.20	17.27	0.32	35.21	10.01	0.14	10.17	0.26	11.00	0.16	**31.74***
DRB					**+10.93 %**							**+0**
5	19.89	0.28	18.43	0.44	39.03	10.65	0.18	11.34	0.32	12.08	0.22	34.80
DRB					**+22.97 %**							**+9.64 %**
	Shih [13]											
3	19.68	0.12	18.22	0.11	38.12	10.43	0.06	11.27	0.09	11.84	0.02	33.72
DRB					**+20.10 %**							**+6.24 %**
5	21.24	0.15	21.59	0.14	43.12	11.02	0.08	12.59	0.11	12.94	0.02	36.77
DRB					**+35.85 %**							**+15.85 %**

4. CONCLUSION

The findings indicated that employing more fractal cells generally lowers total operational costs (training + *PC*s). Although the results indicate that extensive machine inventories do not guarantee good performance, particularly if the company does not already possess them, it was also found that PCs tend to decrease within larger layouts to form virtual cells, thereby reducing product waiting times for machine availability. However, a key trade-off emerged regarding operator qualification: seeking minimal operation times through highly specialized operators may prove uneconomical owing to the significantly higher training costs.

For future work, it is important to analyse the ordering rules to identify the one that best reduces the total cost. Moreover, consider the movement of operators between fractal cells performing operations, and how the lack of availability impacts the total time and cost.

REFERENCES

[1] Gunasekaran, A. (1998). Agile manufacturing: enablers and an implementation framework, *International Journal of Production Research*, Vol. 36, No. 5, 1223-1247, doi:10.1080/002075498193291

[2] Benjaafar, S.; Sheikhzadeh, M. (2000). Design of flexible plant layouts, *IIE Transactions*, Vol. 32, No. 4, 309-322, doi:10.1080/07408170008963909

[3] Saad, S. M.; Lassila, A. M. (2004). Layout design in fractal organizations, *International Journal of Production Research*, Vol. 42, No. 17, 3529-3550, doi:10.1080/00207540410001721790

[4] Askin, R. G.; Ciarallo, F. W.; Lundgren, N. H. (1999). An empirical evaluation of holonic and fractal layouts, *International Journal of Production Research*, Vol. 37, No. 5, 961-978, doi:10.1080/002075499191355

[5] Baykasoglu, A. (2003). Capability-based distributed layout approach for virtual manufacturing cells, *International Journal of Production Research*, Vol. 41, No. 11, 2597-2618, doi:10.1080/0020754031000087229

[6] Venkatadri, U.; Rardin, R. L.; Montreuil, B. (1997). A design methodology for fractal layout organization, *IIE Transactions*, Vol. 29, No. 10, 911-924, doi:10.1023/A:1018540000690

[7] Montreuil, B. (1999). Fractal layout organization for job shop environments, *International Journal of Production Research*, Vol. 37, No. 3, 501-521, doi:10.1080/002075499191643

[8] Shih, Y. C.; Filho, E. V. G. (2014). A design procedure for improving the effectiveness of fractal layouts formation, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 28, No. 1, 1-26, doi:10.1017/S0890060413000474

[9] Shin, M.; Mun, J.; Jung, M. (2009). Self-evolution framework of manufacturing systems based on fractal organization, *Computers & Industrial Engineering*, Vol. 56, No. 3, 1029-1039, doi:10.1016/j.cie.2008.09.014

[10] Aririguzo, J.; Saad, S. M. (2011). Genetic algorithm approach to modelling fractal manufacturing layout, *Proceedings of the 10th International Conference on Modeling and Applied Simulation (MAS 2011)*, 126-135

[11] Jaegler, A. (2018). Fractal organization: preparing for the factory of the future, *IEEE Engineering Management Review*, Vol. 46, No. 4, 136-142, doi:10.1109/EMR.2018.2880438

[12] Elaskari, S.; Venkatadri, U. (2022). Understanding the design continuum between group technology and fractal cell designs for manufacturing systems through the central backup cellular manufacturing system, *Operations Research Forum*, Vol. 3, No. 1, Paper 22, 37 pages, doi:10.1007/s43069-022-00126-y

[13] Shih, Y. C. (2024). Impacts of maintenance policies on fractal layout performance, *International Journal of Simulation Modelling*, Vol. 23, No. 3, 401-411, doi:10.2507/IJSIMM23-3-688

[14] Sandkuhl, K.; Kirikova, M. (2011). Analysing enterprise models from a fractal organisation perspective – potentials and limitations, Johannesson, P.; Krogstie, J.; Opdahl, A. L. (Eds.), *The Practice of Enterprise Modeling (PoEM 2011), Lecture Notes in Business Information Processing*, Vol. 92, Springer, Berlin, 193-207, doi:10.1007/978-3-642-24849-8_15

[15] Peron, M.; Fragapane, G.; Sgarbossa, F.; Kay, M. (2020). Digital facility layout planning, *Sustainability*, Vol. 12, No. 8, Paper 3349, 17 pages, doi:10.3390/su12083349

[16] Vilela, F. F.; Leal, F.; Montevechi, J. A. B.; Piedade, D. D. C. (2020). Effect of human factor performance on the productivity of a manual assembly line, *International Journal of Simulation Modelling*, Vol. 19, No. 3, 365-374, doi:10.2507/IJSIMM19-3-508

[17] Sabioni, R. C.; Daaboul, J.; Le Duigou, J. (2022). Joint optimization of product configuration and process planning in reconfigurable manufacturing systems, *International Journal of Industrial Engineering and Management*, Vol. 13, No. 1, 58-75, doi:10.24867/IJIEM-2022-1-301

[18] Othman, M.; Albar, F.; Bakhsh, A. (2025). The improvement of rewinding workshop using discrete event simulation, *International Journal of Simulation Modelling*, Vol. 24, No. 2, 237-248, doi:10.2507/IJSIMM24-2-721

[19] Zhang, F. L. (2024). Evolutionary algorithm for dynamic resource allocation and its applications, *International Journal of Simulation Modelling*, Vol. 23, No. 3, 531-542, doi:10.2507/IJSIMM23-3-CO14