Topology Planning in Swarm Production System: Framework and Optimization

Akshay Avhad, Casper Schou and Ole Madsen

December 8, 2022

Topology Planning In Swarm Production System: Framework And Optimization

Akshay Avhad[1] Casper Schou[1] and Ole Madsen[1]

Department of Materials and Production
Aalborg University, Fibigerstræde 16, 9220 Aalborg East
akshayra@mp.aau.dk

Abstract. A Swarm Production System (SPS) aims to be an agile and resilient Reconfigurable manufacturing system (RMS) paradigm that incorporates mobile workstations and transport robots on the factory production floor. This paper primarily focuses on SPS's initial but recurring planning stage termed topology planning, which dynamically changes throughout the production runtime with spatially adaptive workstations and transporters handled exclusively by a Topology Manager (TM). TM is essential to multi-variant production with the optimal positioning of the workstations and provides a topology that optimizes the traffic flow for the product carrier robots. TM is a bridge to enable SPS to integrate with general planning and scheduling systems like ERP and MES and is comprised of a Topology Planner (TP) that evaluates the ideal configuration of on factory floor for a batch of product mix and a Reconfiguration Decision System (RDS) that decides on applying the estimated new topology during the batch changeover. The paper proposes a framework for the TM to identify its essential functionalities, responsibilities and working principle in a swarm production system. The paper also describes a grid-based heuristic approach applicable to two-dimensional spatial problems to reduce the complexity of the NP-hard problem. The paper focuses on a framework to estimate a reconfigurable shop floor layout with a Force-directed Graph-theory algorithm. A stochastic statistical model evaluates the performance of the optimal topology for throughput and makespan.

Keywords: Swarm Production System, RMS, Topology Manager, Force-directed, Statistical model

1 Introduction

The concept of Swarm Production System (SPS) proposed in [1] is a more flexible production flow concept compared to the known manufacturing paradigms like Assembly line and Matrix production. In an SPS, the workstations and product conveyances between stations are mobile entities, which can be placed in any location suitably. The main objective is to improve responsiveness to the market's need for producing batches of different of product variants (PV). Each PV has an optimal workstation layout on the shop floor, allowing cost-efficient volume production. The cost for producing a PV largely depends on resource allocation of workstations and part carrying robots and the cumulative inter-workstations travel length. The efficient SPS planning is pivoted on optimal travel cost in a shop floor layout as resource allocation is a task for ERP and MES enterprise systems. The scope of any production system spans Planning, Scheduling and Control; it applies to SPS too.

The planning stage in a production system starts with identifying resources such as machines, actuators, sensors, and workforce for the assembly operation. An assembly or process workstation is an entity that hosts most of these resources as a unit, termed a Workstation robot [WR] in SPS. Every WR has links to other WRs; a chain of these links forms sequences that enables a PV production. A link indicates a direction of material flow carried by a product carrying Transfer Robot [TR]. In the scope of an SPS, a WR can have multiple linkages depending on the number of PV in production, forming a graph structure with x and y positions for the node WR and edges representing the linkages between WRs. These graphs are a topological structure that enables SPS to produce a given set of PVs in a batch with efficiency in process execution. Each batch has an optimal topology. The optimal topology mandates time efficiency in batch production with better throughput and cycle time. The topology also lays a foundation for subsequent scheduling and control activity in SPS with enumeration and task allocations for TRs in sequential assembly operation for a PV. Identifying optimal positions for every WR in the topology is a combinatorial NP-hard problem with factorial time complexity. Therefore, a multi-step optimisation is proposed in the topology estimation problem in SPS. As shown in the figure 1, a Topology Manager (TM) handles the planning stage of identifying, estimating, and optimising the topology in SPS. External ERP and MES systems provide high-level planning and scheduling information essential to the TM's initialisation. Furthermore, the SPS contains a Swarm manager, which executes process level tasks on WRs and TRs.

This paper presents a framework for TM to orchestrate the topology planning, initiating the production process with the transfer of batch information from ERP and MES in the factory and ending with a local optimum topology for production.

2 State of the Art

Operations research has extensively studied the factory layout problem (FLP) associated with optimally localising manufacturing facilities to reduce cost. The nature of the problem of SPS topology planning has similarities with the FLP with the placement of WR on a shop floor.

2.1 FLP in Changeable Manufacturing

SPS is an applied case of conceptual RMS with a practical production philosophy involving autonomous WR and TR entities on the shop floor. The most

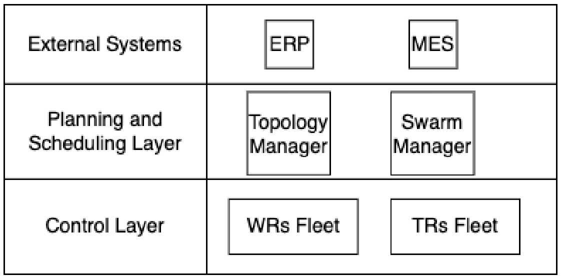


Fig. 1: SPS system level context[1]

practical research questions in the reconfigurable scenario is around addressing adaptability and scalability of transportation within production and deploying the dynamic layout over two dimensional plane [2]. Rosenblatt [3] first addressed the problem of dynamic plant layout (DLP). Heuristic-based solutions and dynamic programming techniques are proposed in [3], and [4] for DLP. Kusiak and Heragu [5] found that heuristics yield near-optimal, computational light solutions relevant to non-uniform spatial vacancies in Flexible Manufacturing Systems (FMS). A hybrid genetic algorithm has been implemented in [6] to optimise continuously change layout requirements in RMS. The effectiveness of this algorithm prevails over the standard genetic algorithms due to its broader search spaces capability. Metaheuristic techniques combined with deterministic ML algorithms are efficient in solving combinatiorial optimization problem in Changeable Manufacturing Systems [7]. The selection of optimal factory configuration is critical to quantifying machines, equipment, robots, and task assignments to all these entities. Line-less Mobile Assembly System (LMAS) [8], is a flexible production paradigm, incorporates a statistical assessment model created in [9] for the early planning stage; in comparison to a discrete event simulation (DES) model that is cumbersome to build in the absence of a suitable scheduler.A cost and time-driven dual approach [10] is proposed for task and location assignment in the planning of LMAS. Therefore, dividing shop floor area into a grid with uniform squares to reduce the time and computational complexities of estimating discrete x and y parameters instead of continuous ones.

2.2 Optimization methods in FLP

Multiple FLP design problems in [11] and [12] are addressed with Mixed Integer Linear Programming (MILP) optimization methods. In [13] a branch and bound approach coupled with MILP is ineffective in solving a large-size problem, while metaheuristic algorithms fare better in comparison. MILP-based solvers shown to have exponential time complexity from medium to large grids impart low practicality in real-world FLP problems in [14]. Hybrid metaheuristic-based experiments performed in [15] for Capacity-based FLP optimally locate factories with the demand such that overall cost due to operation and product transportation is minimal. Sets of metaheuristic solutions like Simulated Annealing (SA) and Genetic Algorithm (GA) are applied to the dynamic reconfiguration of factories in [16][17], and [18]. Quadratic assignment problem (QAP) in [19] and [20] addresses peculiar FLP where the cost is cumulative of distances between facilities and number flows between them. Various QAPs have been used in cross-disciplinary planning facilities for hospitals, supermarkets, and also in precision demanding electronic circuits design are presented in [19] [20] and [21].

2.3 Optimization with Graph theory

A two-stage cost optimisation model in [22] applies graph theory for evaluating the initial solution based on shortest path constraints followed by a selection of more optimal configurations from the first stage. The solution to FLP discussed in [23] and [24] could be modelled as an optimal location solution for vertices of a graph on 2D space with the edge weight representing cost. The spatial layout is optimised, transforming the supergraph into a subgraph that retains the parent's logical edge connections into a topological form, eventually marking it as a graph theory problem in [25][19].

2.4 Optimal approach in Topology Manager(TM)

Most factory layout problems are centred around static location planning, and numerous reasonable heuristic solutions can be derived from them in SPS to perform the assembly operation of a batch mix. The cost and time-tradeoff are essential factors while planning SPS; therefore, a near-optimal approach in an ideal time frame is the best possible solution for a TM. Standalone MILP and metaheuristics are not enough to tackle the large search space needed in identifying topologies on a two-dimensional plane as it indefinitely increases the time and computational complexities as mentioned in [26][27] [28].

The linear increase in topology size with WRs exponentially increases the search space in planning the optimal topology. Most heuristic solutions are deployed in a constrained-based scenario and a fairly static location planning objective. SPS needs a solution capable of identifying dynamic near-optimal topology in a viable time frame. Hence in the following, we will first propose several concepts in a TM. An example follows this at a practical implementation of TM, uncovering some of the abovementioned issues.

3 Topology Manager Framework

The outline of a TM framework is shown in the figure 2. The TM framework is based on utilising the existing enterprise software infrastructure of ERP and MES. The prerequisite to the TM planning process includes a PV list from products scheduled over the next day and an enumerated sequence of WRs for every PV in the list. The pre-planning database hosts the prerequisite data that Batch Processing Module (BPM) later retrieves and processes into batch data. A batch data model is an aggregated data structure for the number of PVs in daily production and their WRs sequences. As seen in the figure 2 the output of the BPM is a logical topology comprising sets of WRs and their connections for every PVs in the batch without spatial information. The Topology estimation and optimization module (TEOM) in the figure 2 identifies a spatial topology based on the logical topology from BPM and optimize before the start of the production of the batch. The TEOM identifies different topologies relevant to the nature of the batch data. Different methods based on graph theory and heuristics are applied in TEOM to identify and optimize the search space for the near-optimal topologies. The set of topologies is the outcome of the TEOM and is forwarded to the Reconfiguration Decision module (RDM) for the decisive deployment of the most optimal topology on the production floor. The last module in figure 2 is an RDM, an inference engine for the selection of the most optimal spatial topologies and decision on the changeover from the currently deployed topology. The changeover process implies a temporal loss in production due to downtime. A reconfiguration process is triggered only if the sum of reconfiguration time loss and the production time with the new topology is less than the production time with the existing topology. In short, the reconfiguration process is skipped when the existing topology fares better, considering the reconfiguration cost. The production topology database has the final plan to be ready for SPS runtime production.

4 Exemplification

The TM represented a generic framework for macro-level planning inside an SPS. Practical implementation requires defined data structure, methods and algorithms in every stage of the TM. The problem TM tries to achieve is a most optimal topology that estimates locations for WRs and the material flow enabled by the topology. We do not consider the processes on WRs nor their flexibility and redundancy, assuming a single process per workstation. Thus the planning goal becomes to optimise the material flow and thereby the distance between WRs and the potential congestion between TRs.

4.1 Batch Processing Module

In figure 3 a multi phase process depicting data flow from ERP and MES to a structured, logical topology representing an SPS batch is shown. A logical topology is an undirected graph data structure with nodes representing a set of WRs for all PVs in a batch and the linkages between WRs as the edges. Phase 1 describes a database cluster hosting a separate schema for daily production PVs and WRs sequence data for each PV. Phase 2 is the interface between TM and the pre-planning database retrieving the PVs list and WRs schema. Lists of WRs are extracted from the phase 2 data depending on the PVs in the

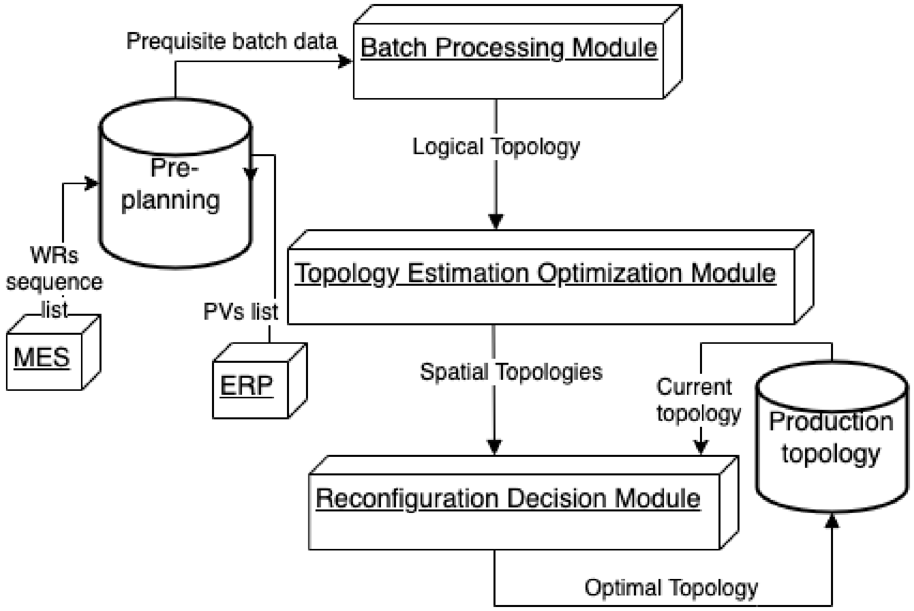


Fig. 2: Topology Manager Framework

production list and aligned in the same order as PVs in the production list, and phase 3 represents this sorted 2D list data form, also known as batch data. The terminology PVs is replaced with Product Instance (PI) in phase 3, indicating that the PV is a product template before being enlisted in a batch. At the same time, PI is the physical entity associated with the PV in production. The batch data from phase 3 is converted to a logical 2D graph topology denoted by G = {V, E} where V represents a set of WRs nodes and edges E retaining the information from the WRs sequences.

4.2 Topology Estimation and Optimization Module

Different graph theory-based approaches are undertaken in TEOM to generate an optimal topology from the input logical topology. The logical topology represents a graph for a complete batch, while every PI in a batch is a subgraph of batch topology.

Objective function SPS differs significantly from a conventional production philosophy; hence, it is at the preliminary stage to understand the cost required to produce a PV. Since the planning stage demands topologies relevant to a batch of PVs, travel distances between the WRs influence the makespan and are hence used as a cost function. Throughout the estimation and optimization process, objective function is based on cumulative eucliedean distances between

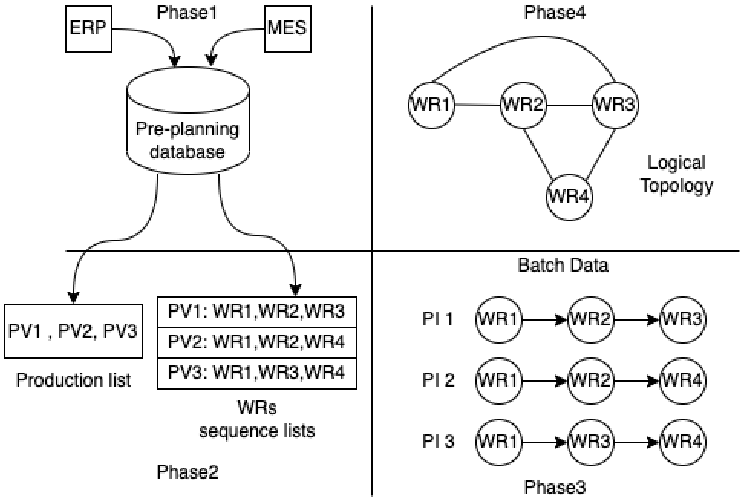


Fig. 3: Formation of Batch and Logical Topology

WRs in PI subgraphs. As mentioned in equation 2 the cost of a topology is the total travel distance required to visit every WR in a sequence for each PI in a batch. The objective function for estimation and optimization stage is minimal travel cost for the complete batch as stated in equation 1

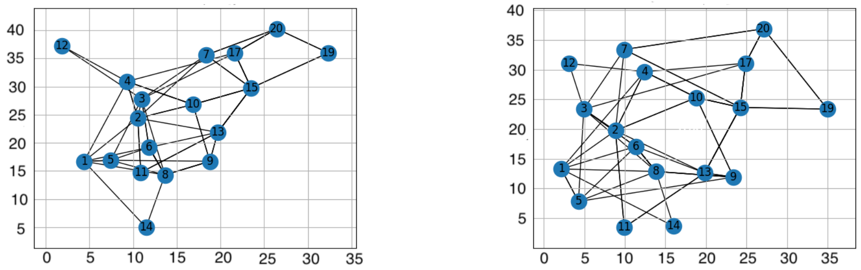
$$C = min \sum_{v=1}^{V} T_v \qquad (1) \qquad\qquad T_v = \sum_{i=1}^{n-1} d(i, i+1) \qquad (2)$$

where, C = Cost of a batch topology
T_v = Travel cost in a PI subgraph
v = PI enumeration in batch
i = WRs enumeration in PI subgraph
n = Total WRs in a PI
d = euclidean distance between WRs

Factory Planning with Logical Topology The placement of workstation nodes on the shop floor is the layout deployment to enable a batch of multiple PVs. Ideally, the workstations could be placed at the closest possible locations to minimize the conveyance time of the product after every process cycle on individual WRs. The distance between the WRs is constrained by factors like minimum spacing required for TR navigation and structural constraints (safety and environmental blockages). Therefore, WR nodes cannot be placed in an overlapped topological configuration even if it establishes a global minimum cost for production but unrealistic in a physical scenario. The Logical topology provides the connected planar graph, and every WRs require spatial coordinates based on the minimal cost function in equation 1.

Estimation and Optimization with Spring topology The freedom in placement of the WR nodes from logical topology increases the complexity of the problem. The edges represents a preliminary path between the nodes which can be redefined in the later stage of path planning for TRs. Physical analogy embedded in a graph with every edge as a spring force that attracts the connecting nodes in the logical topology provides effective heuristic handling for undirected graphs [29] [30].

A graph layout algorithm for drawing positions on a plane known as Force-directed Placement (FDP)[31] layout injects a repelling spring force among the nodes while expanding and contracting the edges in the whole process. FDP tries to draw positions based on the principles of uniform nodes distribution, minimal edge crossings, and uniform edge length but does not guarantee the implementation of these principles in the final layout[31][32]. An implementation is done using Networkx python API spring layout[33] that uses the FDP algorithm to draw the position on a logical topology without any spatial information. The implicit parameters to this API are the repulsive force (k) value and a number of iterations (ITR) determining the node separation on a planar surface and the maximum iterations required to draw the graph, respectively. Topologies based on Networkx spring are displayed in figure 4 for different values of K, illustrating swelling of the topology as the repulsive forces with increasing value of K.



(a) k = 0.70

(b) k =1.30

Fig. 4: Spring FDP topologies

4.3 Reconfiguration Decision Module

The RDM is the inference engine for selecting the topology with the least production cost in equation 1 from the set of spatial topologies generated by the

TEOM. Since the topology is optimized over the cumulative travel cost of every PV in a batch, an additional layer of a performance assessment model is required to evaluate the potential of topology in terms of production KPIs, e.g. throughput and cycle time. The edges in the topology in the SPS planning stage indicate job routing paths between the WRs. Therefore, the stochastic losses due to congestion causing time delay in TRs on overlapping edges are prominent in the Spring topology. A dynamic Discrete Event Simulation (DES) model provides a test-bed for testing scheduling algorithms, which eventually predicts SPS's PI and batch-specific KPIs. Such a DES model is not built yet for an SPS. Hence, in the absence of a suitable scheduler for SPS, a statistical model is described in the equations below with integer-valued uniform distribution stochastic variable X representing the number of occurrences of congestion. The dispatch time of the final product from v^{th} PI from the first WRs in a topology indicates the vacancy for the next PI loading. The product leaving the first WR is dependent on the cumulative process times of subsequent workstations and the stochastic time losses during the TR's conveyance and therefore, it can be written as

$$DT_v = P_1 + \sum_{i=2}^{n} (P_i + I \cdot X) \tag{3}$$

where P_i is a process time on workstation with range [1,n] and X represents uniform distribution stochastic in range [1,x] with a unit time loss of I on every crossing. The start time of a PI depends on the dispatch time of all quantities from the previous PI and its start time as seen in equation 4.

$$ST_v = \begin{cases} 0, & \text{if } v = 1 \\ ST_(v-1) + DT_(v-1), & \text{otherwise} \end{cases} \tag{4}$$

The end timestamp ET_v of vth PI is evaluated in equation 5.

$$ET_v = ST_v + t_v \tag{5}$$

where, t_v represents total makespan for the v^{th} PI of quantity Q_v and stated below. The end times stamp ET_V of the last product V for PI in the batch provides the total required batch production time as seen in equation 6.

$$BT = ET_V \tag{6}$$

Equation 7 is for calculation of makespan for each PI with quantity Q and estimated throughput λ.

$$t_v = \frac{Q_v}{\lambda_v} \tag{7}$$

where, Q_v is total quantity to be produced for v^{th} PI and λ is the throughput for v^{th} PI .Throughput calculation in equation 8 is based on cycle time for 1st product shown in equation 9 and cycle time for later products shown in equation 10.

$$\lambda_v = \frac{1}{CT_1 v + CT_2 v} \tag{8}$$

$$CT_1v = \sum_{i=1}^{n} P_i + \frac{T_v}{S_v} + I \cdot X \tag{9}$$

$$CT_2v = min(TT_v) + I \cdot X \tag{10}$$

where, T_v denotes total travel cost in v^{th} PI subgraph topology scaled by TRs at a speed S_v for that PI and TT_v is an expeted takt-time for vth PI. In the final stages a local optimum topology OT is selected with minimum batch production time $BTmin$ from equation 11.

$$OT = BTmin \tag{11}$$

The reconfiguration depends on the performances of the newly found near-optimal topology against the currently deployed topology. Reconfiguration in equation 12 is performed only when the sum of the changeover span and the batch production time does not exceed the makespan with the current deployed topology $BTcurr$.

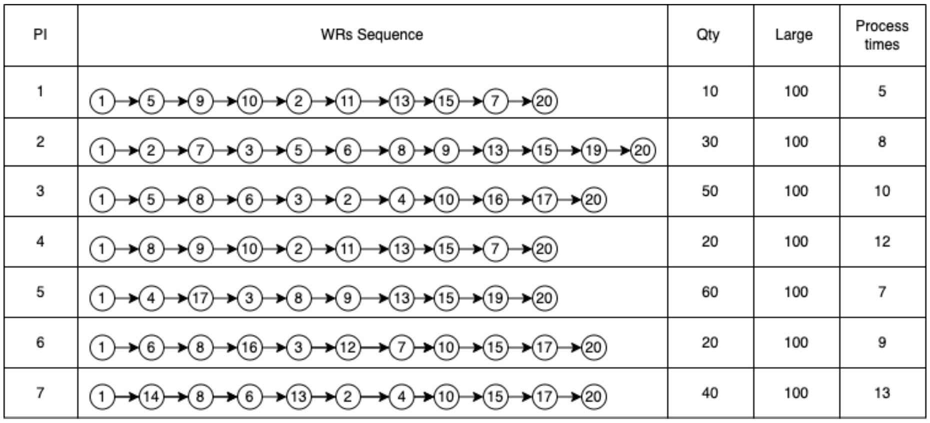
$$RCF = \begin{cases} 1, & \text{if } RT + BT < BTcurr \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

where, RT denotes required reconfiguration time for a new topology for a batch with BT production time or makespan.

5 Experimental Results

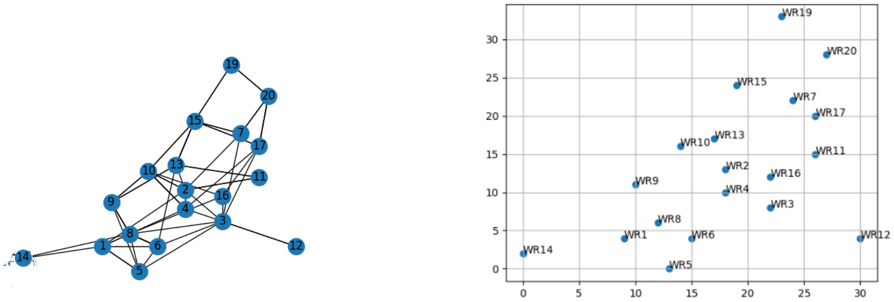
This section describes the numerical implementation performed in Python for the TM. A test batch with seven PIs were used in the numerical exemplification with quantities and sequences illustrated in figure 5. Each PI has two different quantities for standard and larger batch experiments. Uniform process times are assigned to all the WRs in a PI, with every PI having unique process times, as shown in the figure 5. The BPM generates a logical topology of the test batch and feeds it to the TEOM. The population of Spring layouts are generated in the TEOM's topology estimation stage with K value from 1.2 to 2.0 with a step increase of 0.2, and ITR value from 0 to 45 with a step increase of 5.0. The process continues until the cost function converges on the objective function mentioned in the equation 1. The best candidate from the population of Spring topology is found at values K at 1.3 and ITR at 40. The best Spring topology with minimal cost is displayed in the figure 6a. At the same time, the crossings were found on subgraphs for PIs 2,3 and 4 to be 2, 1 and 3, respectively. The variance of the discrete stochastic variable X in equation 3 depends on the number of crossings. The random.randit API generates the integer stochastic variable with a lower limit of 0 and a higher limit as the total number of crossings for respective PIs. A grid-based FLP solution based on the optimal Spring topology in 6b illustrates the two-dimensional spatial positions for WR nodes in the test batch.

The optimal Spring topologies is subjected to performance evaluation through a statistical model from section 4.3 in RDM. The results in figure 7 are generated for smaller batch sizes and a relatively large batch size with different process times and quantities for individual PIs displayed in the legend of the individual figures. From the figure 7a, Spring topology takes 2421 unit time to finish the batch production as compared to 6836 unit time for a larger batch seen in figure 7b.



PI	WRs Sequence	Qty	Large	Process times
1	1 → 5 → 9 → 10 → 2 → 11 → 13 → 15 → 7 → 20	10	100	5
2	1 → 2 → 7 → 3 → 5 → 6 → 8 → 9 → 13 → 15 → 19 → 20	30	100	8
3	1 → 5 → 8 → 6 → 3 → 2 → 4 → 10 → 16 → 17 → 20	50	100	10
4	1 → 8 → 9 → 10 → 2 → 11 → 13 → 15 → 7 → 20	20	100	12
5	1 → 4 → 17 → 3 → 8 → 9 → 13 → 15 → 19 → 20	60	100	7
6	1 → 6 → 8 → 16 → 3 → 12 → 7 → 10 → 15 → 17 → 20	20	100	9
7	1 → 14 → 8 → 6 → 13 → 2 → 4 → 10 → 15 → 17 → 20	40	100	13

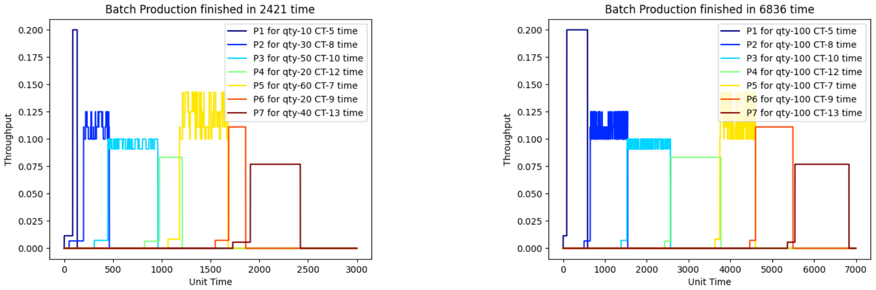
Fig. 5: Batch for numerical exemplification



(a) $k = 1.3$ and $ITR = 40$

(b) Shop floor layout based on Spring

Fig. 6: Optimal Topology from TEOM



(a) Smaller batch

(b) Large batch

Fig. 7: Performances of the Spring Topology

6 Discussion and Conclusion

A flexible and reconfigurable paradigm like SPS enables and requires continuous adaptation to a varying batch mix of product variants. A model-based TM framework presented can achieve a strategic planning objective in SPSs by yielding adaptable topologies to the changing batch mix. The development of the TM is pivoted on integration with the existing manufacturing software ecosystem and extending the capability of an enterprise to plan factory layout for a SPS. The overall goal of the TM is to identify the best possible topology in a defined search space and decide whether to change the new topology or keep the existing one. The reconfiguration process will require an SM that executes task for WRs and TRs on the shop floor.

A near-optimal heuristic approach with graph theory is more computationally viable than a global optimization method as the major challenge is deploying solutions in a short span. Later exemplified with graph theory FDP based optimization with Spring Topology, and a statistical model in RDM. The grid-based approach reduces the computational requirement by optimising discrete space coordinates compared to the continuous ones. The performance KPIs of the topologies are evaluated in TEOM with a mathematical stochastic model in the absence of a suitable simulation tool. The results are outcome of a methodology that assesses planned topologies for the potential performance before being deployed on the shop floor. SPS planning objective can be associated with multiple isomorphic topological graphs apart from the Spring topology. A non-overlapping edge topology is capable of avoiding stochastic losses due to collision and therefore, an extensive study is required in this direction to improve the SPS planning.

Conventional FLP methods are focused on static layout and suited to a defined set of PVs; on the contrary, graph-based TM provides a faster delivery of topological layouts that can be adapted to a batch of changing PVs mix. TM provides a holistic framework that can support relevant graph-based optimization methods apart from Spring topology with an approximated assessment of the performances of the planning stage. TM also represents a digital twin for SPS planning capable of data modelling, optimization methods and decision making for deploying optimal configuration on the production floor.

Due to congestion, the stochastic nature of losses during material flow is subjected to efficient path planning for TRs in SPS. These uncertainties can be reduced with a topological form that enables a collision-free path for TRs in every possible sequence of PVs in a batch. A comprehensive graph theory-based method shall assist explore topologies to deploy shop floor layouts that can lead to a predictive performance assessment. In the future, a grid-based shop floor design, when incorporated with spatial constraints like safe passages, structural blockages, and no deployment zones, enables a pragmatic planning solution in a real-world factory scenario. Furthermore, solutions requiring more expansive search space can be yielded if applied with metaheuristic algorithms, especially in an upscaled production environment with multiple good solutions.

Acknowledgments

This research is supported by the Manufacturing Academy of Denmark (MADE Fast) research program.

List of symbols and abbreviations

λ_v	Throughput for vth PI
BT	Makespan for whole batch
$BTcurr$	Makespan for deployed topology
C	Cost function in Topology estimation and optimization
CT_1v	CycleTime for 1st product in vth PI
CT_2v	CycleTime for all product except the 1st in vth PI
DT_v	Dispatch timestamp of last product on 1st WR for a vth PI
ET_v	End timestamp for a vth PI
I	unit time loss due to congestion on crossings
n	number of WRs in a PI
P_i	Process time of a ith WR, where i ϵ [1,n]
Q_v	Quantity to be produced for a PI in production
RT	Reconfiguration span for topology
S_v	Speed of TR for a PV in production
ST_v	Start timestamp for a vth PI
T_v	Total travel length of PV subgraph
t_v	Makespan for a vth PI
TT_v	Expected Takt-time for a vth PI
V	maximum number of PV in a batch
v	PV number in a batch, where v ϵ [1,V]
X	Stochastic variable with Uniform distribution where X ϵ [1,x]
x	Maximum number of crossings on PV subgraph
BPM	Batch Processing Module
DES	Discrete Event Simulation
DLP	Dynamic Plant Layout
ERP	Enterprise Resource Planning
FDP	Force-Directed Placement
FLP	Factory location Problem
GA	Genetic Algorithm
LMAS	Line-less mobile assembly system
MES	Manufacturing Execution System
MILP	Mixed Integer Linear Programming
PI	Product Instance
PV	Product Variant
QAP	Quadratic assignment problem
RDM	Reconfiguration Decision Module
SPS	Swarm Production System
TEOM	Topology Estimation and Optimization Module
TM	Topology Manager
TR	Transfer Robot
WR	Workstation Robot

References

1. Casper Schou, Akshay Avhad, Simon Bøgh, and Ole Madsen. Towards the Swarm Production Paradigm. In *Lecture Notes in Mechanical Engineering*, pages 105–112. Springer Science and Business Media Deutschland GmbH, 2021.
2. Marco Bortolini, Francesco Gabriele Galizia, and Cristina Mora. Reconfigurable manufacturing systems: Literature review and research trend. *Journal of manufacturing systems*, 49:93–106, 2018.
3. Meir J. Rosenblatt. The Dynamics of Plant Layout. *http://dx.doi.org/10.1287/mnsc.32.1.76*, 32(1):76–86, jan 1986.
4. Daniel G. Conway and M. A. Venkataramanan. Genetic search and the dynamic facility layout problem. *Computers Operations Research*, 21(8):955–960, oct 1994.
5. Andrew Kusiak and Sunderesh S. Heragu. The facility layout problem. *European Journal of Operational Research*, 29(3):229–251, 1987.
6. Xiaoxiao Wei, Sicong Yuan, and Yuanqin Ye. Optimizing facility layout planning for reconfigurable manufacturing system based on chaos genetic algorithm. *http://mc.manuscriptcentral.com/tpmr*, 7(1):109–124, jan 2019.
7. C. Renzi, F. Leali, M. Cavazzuti, and A. O. Andrisano. A review on artificial intelligence applications to the optimal design of dedicated and reconfigurable manufacturing systems. *The International Journal of Advanced Manufacturing Technology 2014 72:1*, 72(1):403–418, feb 2014.
8. Hüttemann G, Buckhorst A, and Schmitt R. Modelling and Assessing Line-less Mobile Assembly Systems. *52nd CIRP Conference on Manufacturing Systems (CMS)*, 81:1–1452, 2019.
9. Robert H. Schmitt, Guido Hüttemann, and Sören Münker. A priori performance assessment of line-less mobile assembly systems. *CIRP Annals*, jun 2021.
10. Armin F. Buckhorst and Robert H. Schmitt. Multi-staged, multi-objective optimization for operation management in line-less mobile assembly systems (LMAS). In *Procedia CIRP*, volume 93, pages 1121–1126. Elsevier B.V., 2020.
11. André R.S. Amaral. On the exact solution of a facility layout problem. *European Journal of Operational Research*, 173(2):508–518, sep 2006.
12. Sunderesh S. Heragu and Andrew Kusiak. Efficient models for the facility layout problem. *European Journal of Operational Research*, 53(1):1–13, jul 1991.
13. Maghsud Solimanpur and Amir Jafari. Optimal solution for the two-dimensional facility layout problem using a branch-and-bound algorithm. *Computers Industrial Engineering*, 55(3):606–619, oct 2008.
14. Md Noor-E-Alam, Andrew Mah, and John Doucette. Integer linear programming models for grid-based light post location problem. *European Journal of Operational Research*, 222(1):17–30, 2012.
15. Gabriel Souto, Igor Morais, Geraldo Regis Mauri, Glaydston Mattos Ribeiro, and Pedro Henrique González. A hybrid matheuristic for the Two-Stage Capacitated Facility Location problem. *Expert Systems with Applications*, 185:115501, dec 2021.
16. Alan R. McKendall, Jin Shang, and Saravanan Kuppusamy. Simulated annealing heuristics for the dynamic facility layout problem. *Computers and Operations Research*, 33(8):2431–2444, 2006.
17. Jianping Dou, Xianzhong Dai, Zhengda Meng, J Dou, X Dai, and · Z Meng. A GA-based approach for optimizing single-part flow-line configurations of RMS. *J Intell Manuf*, 22:301–317, 2011.
18. M Maniraj, V Pakkirisamy, and R Jeyapaul. An ant colony optimization-based approach for a single-product flow-line reconfigurable manufacturing systems. *J Engineering Manufacture*, 231(7):1229–1236, 2017.

19. Eliane Maria Loiola, Nair Maria Maia de Abreu, Paulo Oswaldo Boaventura-Netto, Peter Hahn, and Tania Querido. A survey for the quadratic assignment problem. *European Journal of Operational Research*, 176(2):657–690, jan 2007.
20. T.C. Koopmans and M.J. Beckmann. Assignment Problems and the Location of Economic Activities Author (s): Tjalling C . Koopmans and Martin Beckmann. *Econometrica*, 25(1):53–76, 1957.
21. A. M. Geoffrion and G. W. Graves. Scheduling Parallel Production Lines with Changeover Costs: Practical Application of a Quadratic Assignment/LP Approach. *https://doi.org/10.1287/opre.24.4.595*, 24(4):595–610, aug 1976.
22. Jianping Dou, Xianzhong Dai, and Zhengda Meng. Graph theory-based approach to optimize single-product flow-line configurations of RMS. *International Journal of Advanced Manufacturing Technology*, 41(9-10):916–931, jun 2009.
23. S. G. Boswell. Discussion A reply to ‘A note on ”Similarity of a new greedy heuristic for facility layout by graph theory to an existing approach”‘. *http://dx.doi.org/10.1080/00207549408956928*, 32(1):235–240, 2007.
24. A review of different approaches to the facility layout problems. *Int J Adv Manuf Technol*, 30:425–433, 2006.
25. Donald L. Kreher and Douglas R Stinson. *Combinatorial Algorithms.* CRC Press, sep 1999.
26. Isabela Maganha, Cristovao Silva, and Luis Miguel D.F. Ferreira. The layout design in reconfigurable manufacturing systems: a literature review. *International Journal of Advanced Manufacturing Technology*, 105(1-4):683–700, 2019.
27. Juan Pablo, Usuga Cadavid, · Samir Lamouri, · Bernard Grabot, · Robert Pellerin, and Arnaud Fortin. Machine learning applied in production planning and control: a state-of-the-art in the era of industry 4.0. *Journal of Intelligent Manufacturing*, 31, 2020.
28. Hichem Haddou Benderbal, Mohammed Dahane, and Lyes Benyoucef. A new robustness index formachines selection in Reconfigurable Manufacturing system. *Proceedings of 2015 International Conference on Industrial Engineering and Systems Management, IEEE IESM 2015*, pages 1019–1026, jan 2016.
29. P. Eades. A heuristic for graph drawing. *Congressus Numerantium*, 42:149–160, 1984.
30. Emden R. Gansner and Stephen C. North. Improved Force-Directed Layouts. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1547:364–373, 1998.
31. Thomas M.J. Fruchterman and Edward M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991.
32. C. Walshaw. A Multilevel Algorithm for Force-Directed Graph Drawing. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1984:171–182, 2001.
33. A A Hagberg, D A Schult, and P J Swart. Exploring network structure, dynamics, and function using NetworkX. *7th Python in Science Conference (SciPy 2008)*, (SciPy):11–15, 2008.