Performance Evaluation of Genetic Algorithm and Particle Swarm Optimization in Off-Site Construction Scheduling

Mizanoor Rahman[1], Sang Hyeok Han[1, 2]

1 Department of Building, Civil and Environmental Engineering, Concordia University, Montréal, QC, Canada

2 Centre for Innovation in Construction and Infrastructure Engineering and Management (CICIEM), Gina Cody School of Engineering and Computer Science, Concordia University, Montréal, QC H3G 1M8, Canada

mizanoor.rahman@mail.concordia.ca, sanghyeok.han@concordia.ca

Abstract –

Off-site construction (OSC) is gaining significant attention due to its promising benefits, including reduced time, cost, and waste, along with improved quality, productivity, and safety. However, the dynamic nature of the production process (i.e., non-typical process time) introduces challenges in OSC production line, such as: (i) bottlenecks: (ii) workstation idle time; and (iii) identification of an optimal production sequence. To leverage the full benefits of OSC, a superior production planning and scheduling optimization method become imperative. Therefore, this paper aims to compare the computational performance of the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) for optimizing OSC production schedule. The methodology consists of the three key steps, including: (i) data analysis; (ii) development of GA and PSO algorithms; (iii) implementation of both GA and PSO in a real-life wall panel production line in Edmonton, Canada. The results reveal that GA outperforms PSO in minimizing project completion time (PCT). Specifically, for 160 wall panels, the PCT using GA is 6112 min, whereas with PSO, it is 6122 min. Conversely, PSO produces results more quickly than GA. For the same set of 160 wall panels, the model runtime is 17.97 sec for GA and 6.0 sec for PSO. The findings of this study offer valuable insights for production managers in selecting the most effective algorithm for optimizing production schedules.

Keywords –

Off-site construction; Genetic algorithm; Particle swarm optimization; Manufacturing; Schedule

1 Introduction

Off-site construction (OSC) is a process in which the building components (e.g., wall, floor, roof) are fabricated in a controlled environment (i.e., factory) and then transport it to the site for assembly. The adoption of OSC is growing, as it is reducing construction time, waste, environmental impact while increase productivity, safety, and quality [1]. However, it is essential to achieve optimal efficiency in the production line, as most of the activities are performed within a factory environment. In practice, the process time for same type of building component (e.g., wall) within the same production line dynamically varies. For instance, in a wall production line, the process time for each wall panel is not uniform due to varying design parameters, such as wall length, height, thickness, number of studs, and the presence of doors and windows. Due to this non-typical process time, the production line encounters various challenges, such as: (i) struggling to identify the optimal production sequence; (ii) facing significant bottlenecks; and (iii) dealing with excessive idle time at workstations. To address these challenges, an optimal production schedule is crucial. Therefore, it is paramount to identify the best optimization algorithm specifically tailored for OSC.

In this regard, limited research has been conducted to identify a superior schedule optimization algorithm, specifically tailored for OSC. For example, Altaf et al. [2] compared the optimization performance of PSO and simulated annealing (SA) in the event of OSC production line and found that PSO outperformed SA. Yazdani et al. [3] combined three metaheuristic algorithms, namely differential evolution (DE), imperialist competitive algorithm (ICA), and GA, to simultaneously minimize the duration and cost of precast production processes. They found that DE provides better result compared to ICA and GA. Lee and Hyun [4] used GA and simulated annealing (SA) to create an optimal production schedule for multiple projects. However, the literature reveals a gap in research regarding the identification of the superior optimization algorithm between GA and PSO, specifically tailored for OSC production lines.

Therefore, the primary objective of this paper is to determine the superior optimization algorithm between GA and PSO for minimizing project completion time in

OSC. This research conducted through three key steps, such as: (i) data analysis; (ii) develop GA and PSO model for minimizing project completion time (PCT) and obtain optimal production sequence; and (iii) compare the computational performance of GA and PSO to determine best optimization algorithm for OSC.

2 Literature Review

This section presents a literature review on GA and PSO, as the primary focus of this paper revolves around these two algorithms.

2.1 Genetic algorithm

Genetic Algorithm (GA) draws inspiration from biological evolution as a search and optimization method. It initiates a pool of potential solutions, assesses their fitness for a specific problem, selects the top performers, and merges their characteristics through crossover, occasionally incorporating mutations. This cyclic procedure persists across numerous generations, striving to approach an optimal or nearly optimal solution. GA is renowned for its adaptability, proving effective in addressing intricate issues with extensive solution spaces, and finds widespread application in the domains of scheduling and optimization. In the literature, extensive research has been conducted on using GA for OSC schedule optimization. For example, Ko and Wang [5], Ko and Wang [6] developed a GA-based multi-objective optimization model to address the flow shop-sequencing issue in the manufacture of precast components (PC) while considering the buffer sizes between production stations. Nassar [7] used GA to develop an optimal schedule for reducing project duration and interruption durations in a linear project. However, this model may not be suited for OSC because the working process of linear projects (e.g., road projects) differs significantly from that of OSC projects (e.g., wood-based wall manufacturing). Fan et al. [8] used GA to find the optimal schedules for repetitive projects. They introduced a soft logic strategy (i.e., sequencing) to minimize project costs. Agrama [9] developed a multi-objective GA to minimize project duration, number of interruptions, and resource levelling for a repetitive project (i.e., a multi-story building). Yazdani et al. [3] combined three metaheuristic algorithms, namely differential evolution (DE), imperialist competitive algorithm (ICA), and GA, to simultaneously minimize the duration and cost of precast production processes. Hyun et al. [10] developed a multi-objective optimization model using NSGA-II to reduce production time and labour costs for a continuous modular unit production line. Zhang et al. [11] developed an NSGA-II model to solve the multi-objective optimization problem in off-site construction (i.e., precast production) schedules by considering the impact of disturbance events such as machine malfunctions, order modifications, and unexpected order insertions.

2.2 Particle Swarm Optimization (PSO)

The Particle Swarm Optimization (PSO) is an optimization algorithm inspired by the coordinated movement of bird flocks in nature. In PSO, a group of particles traverses a search space to identify an optimal solution for a given problem. Each particle adapts its position by considering both its individual experience and the collective experience of its peers, all with the goal of discovering the most favourable solution. Several researches have been conducted on PSO for optimizing OSC and flow shop schedule. For example, Tasgetiren et al. [12] used PSO algorithm for minimizing the makespan and total flow time for the permutation flow shop sequencing problems. Guo et al. [13] developed a modified PSO algorithm for obtaining the optimal production schedule by sequencing the manufacturing process. Koulinas et al. [14] created a PSO-based hyper-heuristic algorithm to solve the resource-constrained project scheduling problem (RCPSP). This hyper-heuristic serves as an upper-level controller for multiple low-level heuristics that navigate the solution space. Altaf et al. [15] combined a DES model with optimization using PSO to generate a more realistic schedule that captures the dynamics of the panel prefabrication process. However, this model may not be ideal for comparing its performance with other algorithms (e.g., GA) because it generates mean PCT. Zhang and Yu [16] developed a planning technique utilizing PSO algorithm to optimize the PC transport process. Hayat et al. [17] introduces the hybridization of the particle swarm optimization with variable neighbourhood search and simulated annealing to tackle permutation flow-shop scheduling problems.

In summary, optimizing a production schedule in OSC is crucial for minimizing overall production costs and ensuring faster project delivery. However, previous studies primarily used either GA or PSO to optimize schedules in OSC. Despite the proficiency of both GA and PSO as optimization algorithms, a comprehensive investigation into determining a superior optimization algorithm specifically for OSC production lines remains unexplored. Accordingly, this paper aims to identify a best performing algorithm between GA and PSO explicitly for OSC.

3 Methodology

To fulfill the objectives of this paper, a research framework is summarized in Figure 1. The framework primarily consists of three procedures, such as: (i) estimating process times from historical data; (ii) developing optimization models, including GA and PSO, specifically suited for OSC; and (iii) implementing and comparing the results of GA and PSO using a real-life wall frame manufacturing factory located in Edmonton, Canada. The process time for each component (e.g., wall panel) at every workstation, the number of workstations, and the number of panels to be produced in a given project are used as input. The criteria of this proposed framework involve one-panel flow (i.e., each workstation can only perform its tasks on one panel at a time) and the sequence (i.e., order) of workstations. In this respect, the optimization models are implemented using Python version 3.11.3. Moreover, the models are run on an Intel® Core™ i7 CPU with a processing speed of 3.40 GHz. Simultaneously, their computational performance in terms of PCT and runtime is recorded.

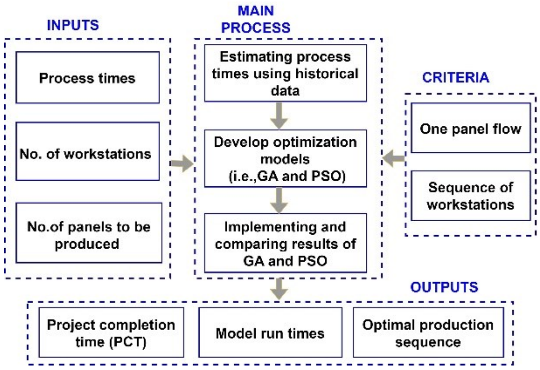

Figure 1. Proposed research framework

3.1 Estimate process times

The overall process to estimate the expected process times for each panel at each workstation using empirical data is shown in Figure 2. In this respect, the process mainly consist of three phases, such as: (i) calculating labor productivity using Equation (1) and (2); (ii) developing histogram to obtain optimistic, most likely and pessimistic productivity because histograms provide a powerful and intuitive way to analyze and interpret the characteristics of a dataset, aiding in decision-making and drawing meaningful conclusions from the data [18]; (iii) calculating weighted average productivity using Equation (6) to incorporate probabilistic process time. It is important to note that before creating the histogram, the data were preprocessed to exclude outliers because outliers significantly influence the determination of realistic statistical parameters, such as mean, upper bound, lower bound[18]. Outliers refer to individual data points that deviate significantly from the overall dataset. Usually represented as individual points beyond the whiskers in a plot. In essence, if any data point is above the upper boundary or below the lower boundary, it is considered an outlier. It can be calculated using Equation (3)-(5). Finally, the process time for each panel at each workstation is estimated in accordance with Equation (2).

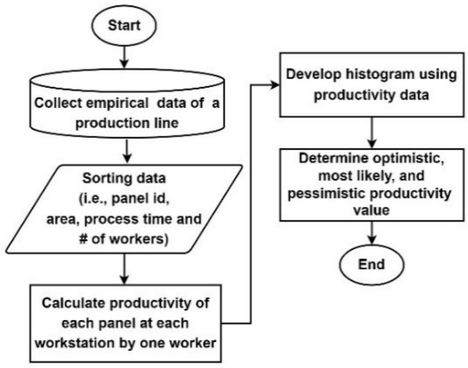

Figure 2. Flow chart for data analysis

$$T = t*N \tag{1}$$

where, T = process time by one worker; t = process time by N number of workers; and N = number of workers.

$$P = A/T \tag{2}$$

where, P = labour productivity; and A = panel area.

$$IQR = Q3\text{-}Q1 \tag{3}$$

where, IQR= interquartile range; $Q1$= first quartile (i.e., median of lower half of data); and $Q3$ = third quartile (i.e., median of upper half of data).

$$\text{Upper bound of data set} = Q3+1.5*IQR \tag{4}$$

$$\text{Lower bound of data set} = Q1\text{-}1.5*IQR \tag{5}$$

The weighted average productivity for each workstation is calculated using the optimistic, most likely, and pessimistic value of productivity using Equation (6).

$$P_{avg} = \frac{(O+4m+P}{6} \tag{6}$$

where P_{avg} = weighted average productivity; O = optimistic productivity; m = most likely productivity; and P = pessimistic productivity.

3.2 Optimization model formulation

The objective of this research is to find the best optimization algorithm between GA and PSO in terms of minimize the project completion time (PCT) and run time in an OSC production line. The objective function is representing in Equation (7).

$$Min\ C\ (Pi,\ Sj) = Min\ [max\{C\ (Pi\text{-}1,\ Sj),\ C\ (Pi,\ Sj\text{-}1)\}+Ti,j\ ] \quad (7)$$

where $C\ (P_i,\ S_j)$ = PCT of ith panel at workstation j; $C\ (P_{i\text{-}1},\ S_j)$ = completion time of (i-1)th panel at workstation j; $C\ (P_i,\ S_{j\text{-}1})$ = completion time of ith panel at workstation (j-1); and $T_{i,j}$ = process time of ith panel at station j. The wall panel production sequence is considered as the decision variables. Moreover, two types of constraints are considered, such as: (i) the wall panel must follow the sequence of workstation; (ii) each workstation should work on a single wall panel (i.e., one panel flow).

3.2.1 GA model

The process flow of minimizing PCT using GA is outlined in Figure 3. The process begins with the initialization of an initial population and each population is evaluated for its fitness based on Equation (7). After that GA generate optimum solution through selection, crossover, and mutation. In the selection process, population with superior fitness are chosen to form a mating pool.

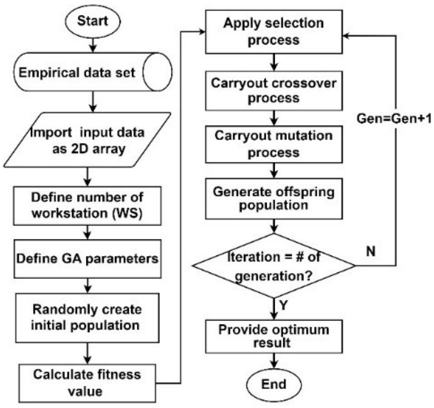

Figure 3. Process flow of GA

In the crossover phase, a partially mapped crossover (PMX) strategy is utilized. This involves generating offspring solutions by selecting a sub-set of gene from one parent and replacing it with another parent. Through this genetic recombination process, offspring can get advantageous attributes from parents, potentially leading to improved solutions. As shown in Figure 4, the random mapping of genes J3-J6 in chromosome 1 (i.e., parent 1) interchange with the genes J6-J1 in chromosome 2 (i.e., parent 2) for crossover, resulting in the generation of offspring 1 (i.e., child 1) and offspring 2 (i.e., child 2).].

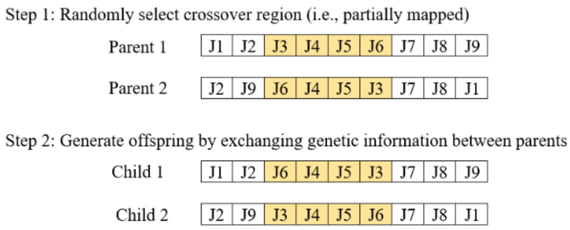

Figure 4. Illustration of crossover process

Furthermore, to preserve population diversity and avoid premature convergence, the mutation operation is employed as an effective strategy. In this study, a two-point swap mutation technique is utilized, where a random pair of genes (i.e., panels) within the chromosome is chosen, and their positions are swapped to generate offspring, forming the foundation for the subsequent generation. As shown in Figure 5, child 1 exchanges the positions of genes J2 and J3, while child 2 swaps the positions of genes J3 and J8, leading to the creation of new offspring.

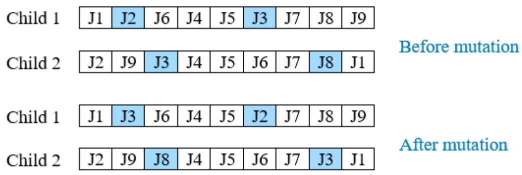

Figure 5. Illustration of mutation process

3.2.2 PSO model

The process flow of PSO is depicted in Figure 6, encompassing essentially six steps, including: (i) randomly generate initial no of particles where each particle represents a solution. For OSC scheduling problem, particle is a list of sequential panels; (ii) calculate the fitness value (i.e., PCT) for each particle; (iii) find local best position (i.e., chose a job sequence that provide minimum PCT between current and previous iteration) and global best position (i.e., chose the job sequence that provides minimum PCT among all the particle in current iteration); (iv) the iteration continue if it does not reached maximum number; (v) update the velocity of the particles as per Equation (8); and (vi) update the position (i.e., sequence of panels) for next iteration.

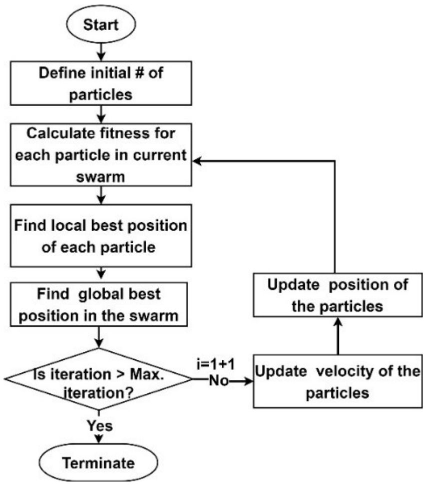

Figure 6. Process flow of the PSO algorithm

$$Vi(t+1) = \omega vi(t)+c_1r_1(p(t)(i,lb)-xi(t)+ c_2r_2(p(t)(gb)-xi(t) \quad (8)$$

where $v_i^{(t+1)}$ = velocity of i[th] particle at (t+1) iteration; ω = inertia weight; $v_i^{(t)}$ = velocity of i[th] particle at current (i.e., t) iteration; c_1 and c_2 = acceleration coefficient; r_1 and r_2 = random numbers value between 0 and 1; $p^{(t)}_{(i,lb)}$ = local best of i[th] particle at current iteration (i.e., t); $p^{(t)}_{(gb)}$ = global best positionProceedings.

4 Implementation and results

The proposed scheduling method is applied to a light gauge steel (LGS) wall panel production line located in Edmonton, Canada, dedicated to manufacturing light gauge steel (LGS) wall panels for both commercial and residential building. The production line primarily consists of three workstations, such as: (i) assembly station where the required number of studs and tracks are prepared based on the shop drawings; (ii) framing station, where the framing components are securely fastened together to form wall panels; and (iii) sheathing station where the drywall is installed on framed components. In this paper, the process time for 166 wall panels is collected and analysed. However, for illustrative purposes, the collected data for 10 panels are shown in Table 2. Subsequently, Equations (1) and (2) are employed to calculate labour productivity for each panel. For example, the area of panel E831 in Table 2 is 94 ft^2, it takes 41 minutes to complete by 2 workers at assembly workstation. Therefore, the process time by one worker is 82 minutes, and labor productivity is calculated as 94/82= 1.15 ft^2/min. Similarly, productivity for panel E831 at the framing and sheathing workstations is 0.94 ft^2/min and 1.57 ft^2/min, respectively.

From the data analysis, it is evident that productivity is not consistent, meaning it varies from panel to panel. To address this variability in productivity, a weighted average productivity is adopted to determine the expected productivity for each workstation. To derive the optimistic, most likely, and pessimistic productivity, histograms are constructed for each workstation using 166 empirical data points. As illustrates in Figure 7 the framing station's diverse productivity metrics, with optimistic, most likely (i.e., median), and pessimistic values of 0.16 ft2/min, 1.15 ft2/min, and 3.19 ft2/min, respectively. Similarly, Figure 8 shows the sheathing station’s productivity statistics, showcasing optimistic, most likely, and pessimistic values of 0.14 ft2/min, 0.79 ft2/min, and 2.29 ft2/min, respectively. As shown in Figure 9 , the nailing station’s productivity spanning 0.14 ft2/min (optimistic), 0.71 ft2/min (average), and 1.91 ft2/min (pessimistic) The weighted average productivity for the assembly, framing, and sheathing workstations is calculated as 1.33 ft2/min, 0.93 ft2/min, and 0.82 ft2/min, respectively. These productivity values are employed to calculate the process time for each panel at each workstation, as presented in Table 1

Table 1. Estimated process time at each workstation

Panel Id	Panel area (ft^2)	Process time (min)		
		Assembly	Framing	Sheathing
E 1099	160	60	57	65
E 819	111	42	40	45
E 561	132	50	47	54
E 767	130	49	46	53
E 779	130	49	46	53
E 861	130	49	46	53
E 1140	136	51	49	56
E 807	136	51	49	55
E 1129	92	35	33	37

Table 2. Sample empirical data for 10 panels

Panel Id	Panel area (ft^2)	Assembly station			Framing station			Sheathing station		
		Process time (min)	No of workers	Productivity (ft^2/min)	Process time (min)	No of workers	Productivity (ft^2/min)	Process time (min)	No of workers	Productivity (ft^2/min)
E 831	94	41	2	1.15	25	4	0.94	20	3	1.57
E 789	151	65	1	2.32	110	3	0.46	100	3	0.50
E 780	132	48	1	2.75	40	3	1.10	85	3	0.52
E 809	56	38	2	0.73	49	3	0.38	24	3	0.78
E 814	109	65	2	0.84	60	3	0.61	72	3	0.50
E 820	62	26	2	1.19	32	3	0.64	54	3	0.38
E 790	22	15	2	0.74	11	2	1.00	13	3	0.57
E 651	136	35	2	1.94	55	2	1.23	47	2	1.44
E 622	69	15	2	2.31	59	2	0.59	33	2	1.05
E 630	45	20	2	1.12	29	2	0.77	40	2	0.56

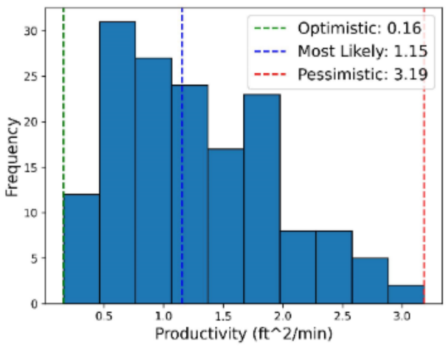

Figure 7. Histogram for assembly station

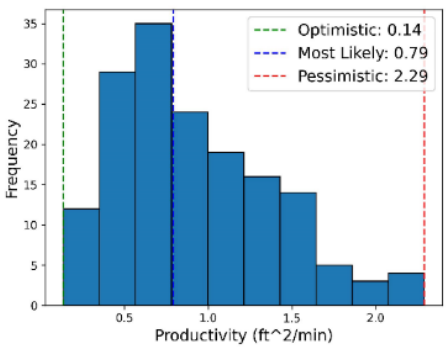

Figure 8. Histogram for framing station

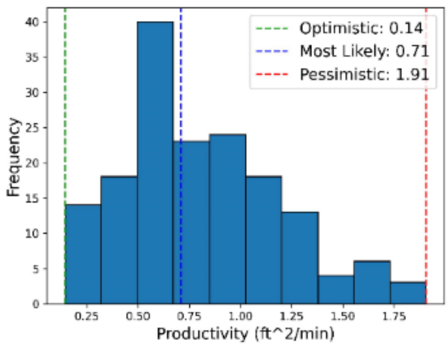

Figure 9. Histogram for sheathing station

4.1 Application of GA

To automate the implementation of GA and enhance the efficiency of minimizing PCT through panel sequencing, a Python-based script has been meticulously developed. The adopted GA parameters are follows: (i) population size 20; (ii) mutation rate 0.1; (iii) crossover rate 0.8; (iv) number of generations 2500; and (v) number of panels 50. As shown in Figure 10, following 1600 iterations, the calculated optimum PCT is 2196 minutes.

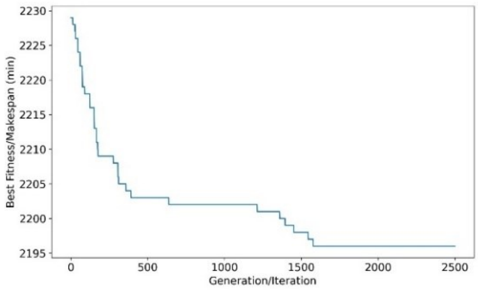

Figure 10. PCT during each iteration using GA

4.2 Implementation of PSO

The objective of this PSO-based optimization model is also to minimize the PCT by sequencing the wall panel production. The selected parameters for running the PSO model are as follows: (i) number of wall panels - 50; (ii) number of particles - 20; (iii) number of iterations - 500; (iv) inertial weight - 0.5; (v) cognitive weight - 2.0; and (vi) social weight - 2.0. As shown in Figure 11, the convergence achieved at 150 iterations and, the PCT reduced from 2237 min to 2226 min.

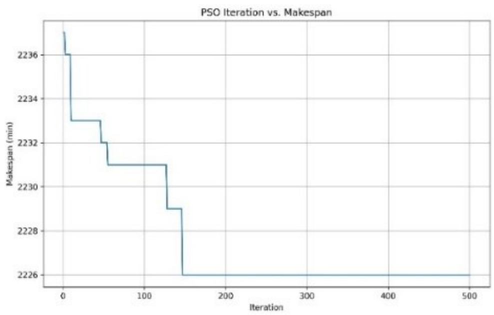

Figure 11. PCT during each iteration using PSO

4.3 Comparison of results

To identify the best-performing algorithm between GA and PSO for minimizing PCT in OSC, 13 sets of wall panels are selected, such as number of wall panel 10,20,30,40,50,60,70,80,90,100,120,140 and 160. As shown in Figure 12, the performance of GA for minimizing PCT is outperformed PSO. For instance, the minimum PCT for 160 panels is 6112 min using GA, whereas with PSO, it is 6122 min. Moreover, the run time at each iteration is record for both GA and PSO. As shown in Figure 13, the run time for GA is relatively higher than PSO. For example, the runtime for 160 panels using GA is recorded as 17.97 sec, whereas with PSO, it is 6.0 sec.

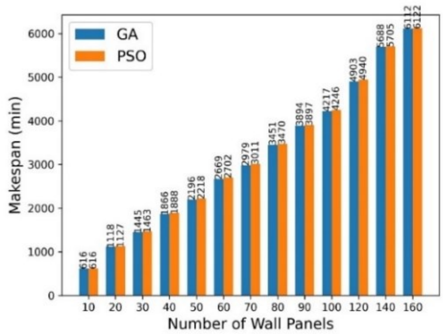

Figure 12. Comparative PCT for GA and PSO

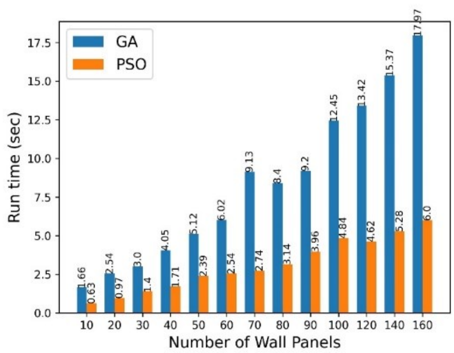

Figure 13. Comparative run time for GA and PSO

5 Conclusions and future works

In this study the performance of GA and PSO for minimizing PCT in OSC production schedule is evaluated. The proposed methodology employed GA, and PSO to optimize OSC production schedule in terms of PCT and run time. In the data analysis stage, a histogram is developed to estimate expected productivity for each workstation. The proposed research framework is implemented on a light gauge steel (LGS) wall panel production line at Edmonton in Canada. Each model (i.e., GA and PSO) is implement in 13 sub-sets of wall panels to minimize the PCT. The results demonstrate that GA provides better results than PSO for minimizing PCT. For instance, the PCT using GA is 6112 min, while it is 6122 min using PSO for 160 wall panels. In contrast, PSO generates output relatively faster than GA for the same set of 160 wall panels. For example, the model run time for GA is around 11.97 sec, while it takes around 6 sec for PSO.

The original contribution of this research assists planners in choosing the best optimization algorithm, eliminating the need for trial and error with multiple algorithms. While this study yields satisfactory outcomes, it is limited to the comparison between two algorithms. To address this limitation, in the future, the proposed methodology can be further expanded by comparing these two algorithms with other search algorithms, including Simulated Annealing, Ant Colony, and Tabu Search algorithms, to find the best performing algorithm for this type of optimization problem, specifically in the context of OSC.

Acknowledgements

The authors gratefully acknowledge the financial support provided by the Natural Sciences and Engineering Research Council of Canada (NSERC) to conduct this research. Additionally, they extend their heartfelt gratitude to Angat Pal Singh Bhatia for generously providing the production line data.

References

[1] Rahman, M. and H.R. Sobuz. Comparative study of IPS & PPVC precast system—A case study of public housing buildings project in Singapore. in Proceedings of the 4th International Conference on Civil Engineering for Sustainable Development (ICCESD 2018), KUET, Khulna, Bangladesh. 2018.

[2] Altaf, M.S., et al., Integrated production planning and control system for a panelized home prefabrication facility using simulation and RFID. Automation in construction, 2018. 85: p. 369-383.

[3] Yazdani, M., et al., Production scheduling of off-site prefabricated construction components considering sequence dependent due dates. Environmental Science and Pollution Research, 2021: p. 1-17.

[4] Lee, J. and H. Hyun, Multiple modular building construction project scheduling using genetic algorithms. Journal of Construction Engineering and Management, 2019. 145(1): p. 04018116.

[5] Ko, C.-H. and S.-F. Wang, Precast production scheduling using multi-objective genetic algorithms. Expert Systems with Applications, 2011. 38(7): p. 8293-8302.

[6] Ko, C.-H. and S.-F. Wang, GA-based decision support systems for precast production planning. Automation in construction, 2010. 19(7): p. 907-916.

[7] Nassar, K., Evolutionary optimization of resource allocation in repetitive construction schedules. Journal of Information Technology in Construction (ITcon), 2011. 10(18): p. 265-273.

[8] Fan, S.-L., K.-S. Sun, and Y.-R. Wang, GA optimization model for repetitive projects with soft logic. Automation in Construction, 2012. 21: p. 253-261.

[9] Agrama, F.A. Multi-objective genetic optimization for scheduling a multi-storey building. in The International Conference on Civil and Architecture Engineering. 2014. Military Technical College.

[10] Hyun, H., et al., Multiobjective optimization for modular unit production lines focusing on crew allocation and production performance. Automation in Construction, 2021. 125: p. 103581.

[11] Zhang, R., et al., Green optimization for precast production rescheduling based on disruption management. Journal of Cleaner Production, 2023. 420: p. 138406.

[12] Tasgetiren, M.F., et al., A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem. European journal of operational research, 2007. 177(3): p. 1930-1947.

[13] Guo, Y., et al., Applications of particle swarm optimisation in integrated process planning and scheduling. Robotics and Computer-Integrated Manufacturing, 2009. 25(2): p. 280-288.

[14] Koulinas, G., L. Kotsikas, and K. Anagnostopoulos, A particle swarm optimization based hyper-heuristic algorithm for the classic resource constrained project scheduling problem. Information Sciences, 2014. 277: p. 680-693.

[15] Altaf, M.S., M. Al-Hussein, and H. Yu. Wood-frame wall panel sequencing based on discrete-event simulation and particle swarm optimization. in ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction. 2014. IAARC Publications.

[16] Zhang, H. and L. Yu, Dynamic transportation planning for prefabricated component supply chain. Engineering, Construction and Architectural Management, 2020. 27(9): p. 2553-2576.

[17] Hayat, I., et al., Hybridization of Particle Swarm Optimization with Variable Neighborhood Search and Simulated Annealing for Improved Handling of the Permutation Flow-Shop Scheduling Problem. Systems, 2023. 11(5): p. 221.

[18] Sheng, Y., et al., Outlier identification in radiation therapy knowledge-based planning: a study of pelvic cases. Medical physics, 2017. 44(11): p. 5617-5626.