Wood-Frame Wall Panel Sequencing Based on Discrete-Event Simulation and Particle Swarm Optimization

Mohammed Sadiq Altaf [a], Mohamed Al-Hussein [a] and Haitao Yu [b]

[a]Hole School of Construction Engineering, University of Alberta, Canada
[b]Landmark Group of Builders Ltd., Canada
Corresponding author email: msaltaf@ualberta.ca

Abstract-

In recent years off-site construction has become popular in North America due to the superior quality of the product, improved productivity, and reduced environmental impact. The panelized construction approach is one of the most readily utilized off-site construction methods. In a wood-frame panelized construction plant, wall panels are customized according to various design parameters such as length; height; number of studs, windows, and doors; panel type; and number of walls. These design parameters affect the processing time at each station in the plant, while the panel sequence affects the waiting time between stations. Due to this dynamic nature of the fabrication process, it is challenging to automatically generate an optimal panel sequence, as a result this task is performed manually in current practice. This paper focuses on integrating discrete-event simulation (DES) with an optimization algorithm in order to automate the panel sequencing process. Processing time at each station is calculated based on a task time formula which is a function of the design parameters of the panel, while delay is calculated based on a distribution derived from historical data. A particle swarm optimization (PSO) algorithm is integrated with the simulation model using a central database in order to generate an optimal panel sequence. The proposed method will eliminate the manual work required for panel sequencing, and is expected to reduce production time up to 10%. The proposed method is implemented in a wood-frame panelized construction plant as a case study.

Keywords-

Panelized construction; Discrete-event simulation; Particle swarm optimization; Panel sequencing

1 Introduction

Off-site construction is increasingly being embraced as the preferred construction approach due to the superior quality of the product, improved productivity, and reduced negative environmental effects. Among the various off-site construction methods, panelized construction is becoming widely utilized due to its design flexibility and lower on-site assembly cost. In this regard, a study by the Building Systems Council (National Association of Home Builders) has revealed that panelized construction reduces waste and construction time compared to traditional stick-built construction [1].

As the majority of the activities in a panelized approach are performed in a factory environment, it is important to achieve optimal productivity in the production line. Among the various assembly line classifications, the panel production line is a mixed-model asynchronous assembly line in which *n* number of jobs (panels) go through *m* number of stations in a series configuration [2]. In a wood-frame panel fabrication plant, wall panels are customized according to various design parameters such as length; height; number of studs, windows, doors; sheets of sheathing, walls; and panel type. These design parameters affect the processing time at each station, while the panel sequence affects the waiting time between stations. As different panels require different processing times at each station, it is important to produce the wall panels in the optimal sequence in order to minimize the maximum completion time of all panels, also known as the “makespan”. This kind of assembly line optimization problem is proven to be a non-deterministic polynomial-time hard (NP-hard) problem [3]. Several studies have been conducted which have sought to optimize the job sequence in a flow shop configuration using lean principles, simulation, genetic algorithm (GA), tabu search, and particle swarm optimization (PSO) [4][5][6][7]. Shewchuk and Guo (2012) have utilized lean to optimize the wall panel stacking and sequencing in residential construction [8]. However, few studies have been carried out to optimize panel production sequence within the home building industry.

Discrete-event simulation (DES) has been used extensively to model flow shop manufacturing processes. Hammad et al. (2002) have developed a simulation model for manufactured housing processes to improve both productivity and the quality of the product [9]. Garza-Reyes et al. (2012) have used DES to find the optimum line balance for every stage in the home production process [10]. DES-based optimization has also been applied to construction and manufacturing in order to find near-optimal solutions. Dengiz and Alabas have used a simulation model together with a tabu search to find the optimum number of kanbans in a just in time (JIT) system [11]. Rezg et al. (2004) have proposed a methodology combining simulation and GA to optimize maintenance and inventory control policies [12]. Lu et al. (2008) have developed an automated resource-constrained critical path analysis using DES and PSO [6]. More recently, Mahdavi et al. (2011) have developed a modified chaotic ant swarm simulation-based optimization model to solve the flexible manufacturing system scheduling problem [7].

As the construction industry is moving towards factory built houses, it is important to sequence the jobs at the plant to improve the productivity by implementing the knowledge of the manufacturing industry. This paper describes a methodology which begins by integrating DES with an optimization algorithm to find the optimal production sequence for a wood-frame panel production line. The formulation of the DES model of the production line and the integration of the model with a PSO algorithm is then presented. Finally, the model results are compared with the current production sequence to measure the effectiveness of this method.

2 Methodology

The methodology is developed using DES-based optimization for the panel production sequence in a flow shop configuration. Wood-frame wall panel information is uploaded into a database from the 3D model. In the optimization environment, the simulation model reads panel information such as length; width; type; and number of studs, windows, doors, and sheets of sheathing from the database and runs the simulation in order to obtain the makespan for all the panels. The panel sequence is then updated in the database based on the optimization algorithm, and the simulation model is run again based on the new sequence. This process continues until the optimal sequence is achieved. The methodology is shown in Figure 1.

The database containing the panel information and sequence is linked with both the simulation model and optimization algorithm. The simulation model reads panel information from the database and calculates the panel processing time in each station. The optimization algorithm updates the panel sequence in the database based on the makespan for all the panels. After finding the optimal panel sequence, the algorithm is terminated, with the optimum sequence and makespan outputted from the model.

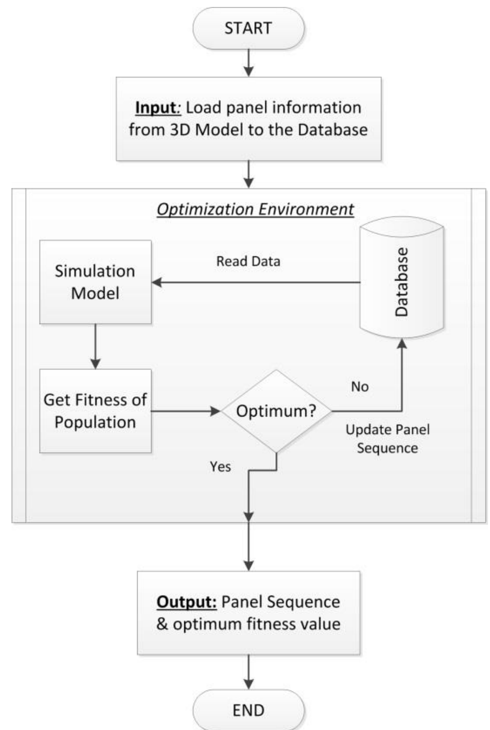

Figure 1. DES-based optimization to find optimal panel sequence.

3 Simulation Model Formulation

The DES-based optimization method is implemented at Landmark Building Solutions (LBS), a wood-frame wall panel fabrication plant located in Edmonton, Alberta, Canada. A simulation model of LBS's wood-frame wall panel production line is developed in Simphony.NET, an integrated environment effective for building simulation models of construction activities [13]. The user builds a simulation model within Simphony by creating instances of modeling elements that resemble real components of a system/process, and linking them together in ways similar to those that exist in a real system. The following sections describe Landmark wall production line system, construction of the task time formula and the simulation logic.

3.1 Landmark Wall Production Line

Landmark Group of Builders, a major production homebuilder in Alberta, Canada, has established a wood-frame panel fabrication plant in east Edmonton, where wood-frame open-wall panels and floor panels are produced and transported to the site for on-site assembly. This manufacturing facility is equipped with state-of-the-art production lines utilizing computer numerical control (CNC) technology which are capable of producing building components for 3 homes in an 8-hour shift.

The wall panel production begins at the framing station, where exterior and interior walls are assembled using CNC machinery. To maximize the utilization of the CNC table, what will ultimately be divided into single-wall panels, which are fabricated as multi-wall panels equal in length to the maximum length of the CNC table (40 feet). From the framing station, the multi-wall panels move to the sheathing station, where the sheathing for the exterior walls is placed by workers at the station and then nailed using another CNC machine, known as the multi-function bridge. All multi-wall panels (both interior and exterior) are also marked with their panel identification number at this station. The exterior multi-wall panels then advance from the sheathing station to the spray booth for application of spray-foam insulation. Interior multi-wall panels, alternatively, advance directly from the sheathing station to the interior wall waiting line. Using a transfer cart, the interior multi-wall panels are also cut into single-wall panels, and then moved to the interior packaging area for shipment. All exterior multi-wall panels are cut into single-wall panels after spraying and moved to the exterior wall waiting line. Exterior panels without windows and doors are moved to the wall magazine line for shipment, while those requiring windows/doors are first moved to the window/door installation station and then on to the wall magazine line.

3.2 Construction of Task Time Formula

In order to simulate the wall production line, a time study is conducted at each station to develop a task time formula by which to calculate the processing time of the panel at every station. The task time formula is developed based on the time needed to perform each task for a given panel and station. Equation (1) shows the developed task time formula for the framing station.

$$\textit{Process time (sec)} = T.B. + S * 9.92 + M * 29.59 + L * 20.58 + W * 77.05 + D * 44.43 + Drill * 5.40 + Cut * 7.10 + Nail \quad (1)$$

Where T.B. is the time needed to place the top and bottom plates (sec); S is the number of single-studs; M is the number of multiple-studs; L is the number of L-studs and double-studs; W is the number of windows; D is the number of doors; Drill is the number of drill holes in the panel; Cut is the number of cut-zones in the panel; and Nail is the Time needed to refill the nails (sec).

The time needed to place top and bottom plates, single- and multiple-studs, and windows and doors at the framing station are observed and recorded. The average time to perform each task is used to generate the above equation. The delay time is also observed and categorized into different delay types. For the framing station, there are primarily four types of delays: machine breakdown, material supply delay, worker away, and error correction. The frequency of each type of delay is observed and the probability of occurrence is calculated. Each type of delay is fitted in a triangular distribution. Similarly, the task time formula is constructed for other stations in the wall production line. The task time formula is validated by comparing the calculated process time with the actual process time.

3.3 Simulation Logic

The wall panel production process is simulated in Simphony.NET, with the simulation flowchart shown in Figure 2. *n* number of multi-wall panels are created as the model entities. All entities (multi-wall panels) go through a “set attribute” function where all the panel attributes, i.e., panel length, width, and type; spray area; number of walls studs, windows, doors, and sheets of sheathing; and panel sequence, are read from the database. Each entity (multi-wall panel) then goes through different stations (tasks) i.e., framing station, sheathing station, spray booth, transfer cart, and window installation station. The task time at every station for each panel is calculated based on the task time formula. The framing station, sheathing station, and transfer cart have one resource each and the spray booth and window installation station have two resources each. Each wall panel has to capture the resource before entering the station and can release the resource after capturing the resource of the following station. If the resource is not available, the panel will wait in the previous station. Following completion of the spray booth task, each exterior multi-panel entity in turn generates multiple entities (single-panel) based on the number of walls in each multi-wall panel. After completing all the tasks, the total production time for each panel is stored and the entity quits the simulation model.

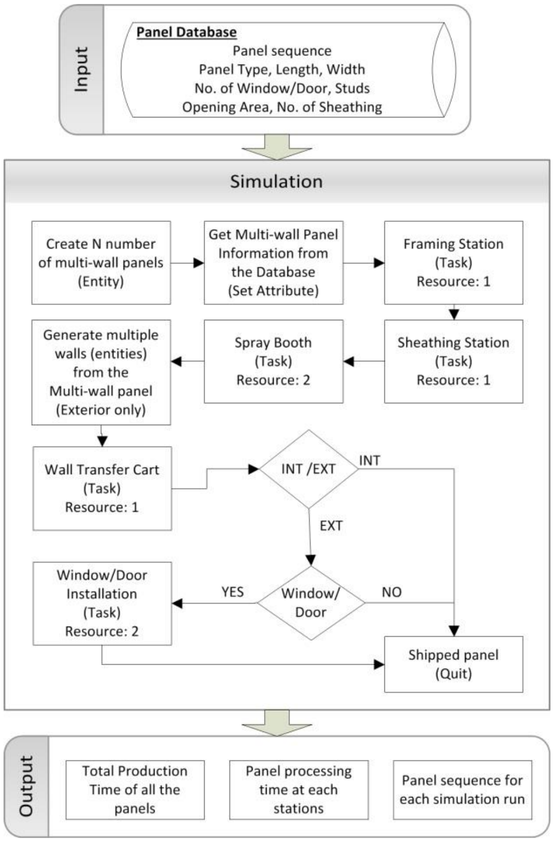

Figure 2. Simulation flowchart

Figure 3 shows the simulation model developed in Simphony. The model consists of several *composite* elements representing different stations. The framing station composite element is shown in Figure 3. The framing station model includes a *capture* element to capture the resource and a *task* element to simulate the process time. Another composite element, *delay*, is used to calculate the delay time. Inside the delay, the probability *branch* is used to capture the likelihood of each type of delay occurring. Following the delay element calculations, the entity moves out of the framing station. Also, in between tasks the model records the time using the *set attribute* element.

4 Particle swarm optimization

PSO is a population-based evolutionary algorithm proposed by Kennedy and Eberhart [14]. In the PSO algorithm, the search is performed by a set of particles (i), and the information is shared between all the particles in order to find the optimal solution. Each particle is considered as a point in a D-dimensional space and has a velocity and position value. The position and velocity values of the i^{th} particle are denoted as $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ and $v_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$, respectively. Each particle moves towards the best solution of the entire swarm (i.e., the "global best") by updating its position and velocity values after every iteration. Initially, the position and velocity values are assigned randomly to each particle in order to start the search. Then the values are updated based on the results of all previous iterations following Equations (2) and (3).

$$ {}_{id}^{k+1} = w * {}_{id}^{k} + c_2 * r_2 * \left(p_{id}^{k} - x_{id}^{k}\right) + c_2 * r_2 * \left(p_{gd}^{k} - x_{id}^{k}\right) \quad (2) $$

$$ x_{id}^{k+1} = x_{id}^{k} + {}_{id}^{k+1} \quad (3) $$

Each particle's best position is represented by p_{ld} and its global best position is represented by P_{gd}. c_1 and c_2 are the cognitive parameter and social parameter, respectively. In this model, both values are set to 2. r_1 and r_2 are random numbers uniformly distributed from 0 to 1. k is the iteration number and w is the weight inertia required to control the impact of the previous velocity value on the current velocity. In this model the value of w is set to 0.9 at the start and then is decremented by a factor of 0.975 after every iteration (i.e., $w^{k+1} = 0.975 * w^k$). The search process is terminated once the maximum number of iterations is reached.

In order to implement the PSO algorithm in a sequencing problem, a heuristic rule called *Smallest Position Value* (SPV) is applied [15]. In this panel sequencing problem, each particle has a continuous set of position values representing every multi-wall panel. If the model is run to optimize the sequence of 50 multi-wall panels, each particle in the PSO search will have 50 position and velocity values representing each multi-wall panel. Since the particle itself cannot represent a sequence, it is determined by the position values ($x_{i1}, x_{i2}, \ldots, x_{iD}$) of the particle ($x_i$). According to the SPV rule, the panel with the smallest position value will be first in the production sequence; the panel with the second-lowest position value will be second in the sequence, and so on. After running the simulation, the position value is updated based on the fitness value, and a new sequence is generated following the SPV rule.

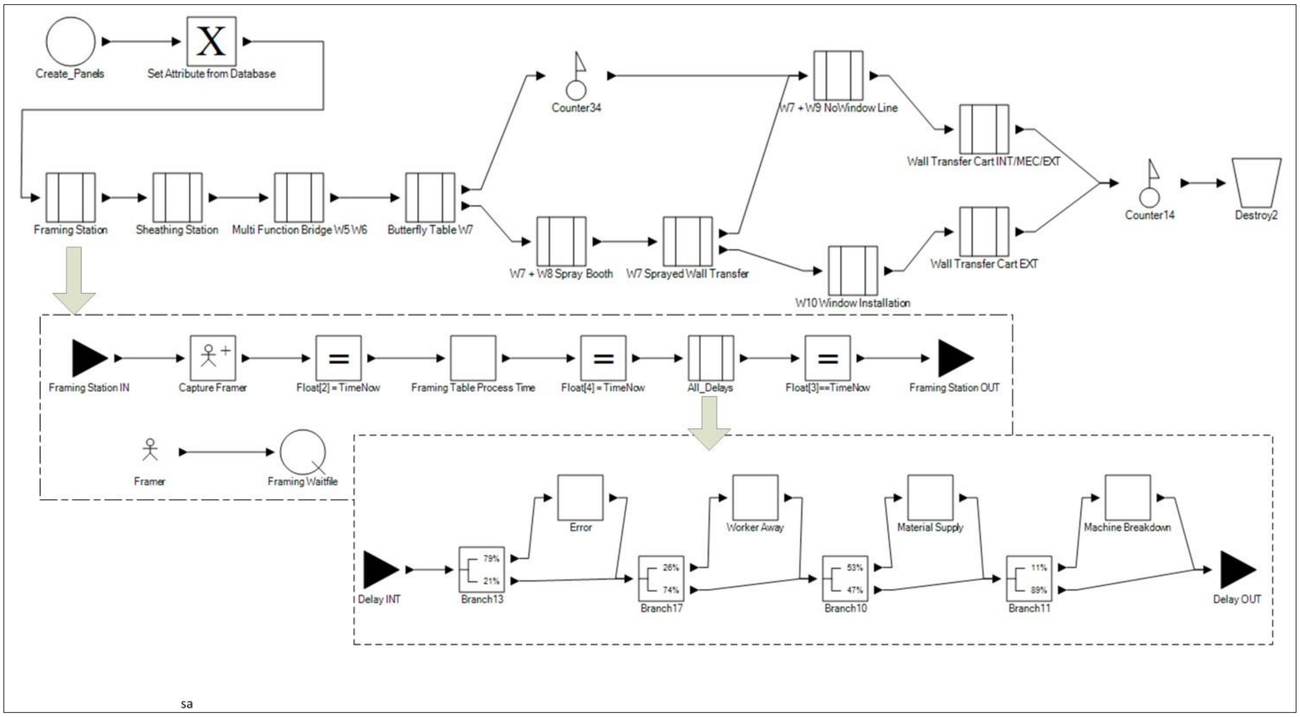

Figure 3: Simulation model in Simphony.NET.

Figure 4 illustrates the integration between the PSO algorithm and the simulation model. The PSO search algorithm's utilization of DES for the panel sequencing problem is summarized as follows:

- Step 1: The user uploads multi-wall panel information (length; width; type; and number of windows, doors, studs, sheets of sheathing, and walls) into the database.
- Step 2: The PSO model assigns initial position and velocity values to every panel under each particle and updates them in the database. The position values are generated randomly between 0.0 and 4.0. Initial velocities are created randomly between -4.0 to 4.0.
- Step 3: The simulation model reads the panel information from the database. The panel is sorted based on the SPV rule, and the simulation model is run for every particle, with the total makespan stored as the personal best value (p_{ld}) for each particle.
- Step 4: The PSO model updates the iteration counter: $k = k + 1$.
- Step 5: The PSO model updates the inertial weight: $w^k = w^{k-1} * \alpha$, where α is decrement factor.
- Step 6: The PSO model updates the velocity of each particle satisfying Equation (2).
- Step 7: The PSO model updates the position of each particle in the database satisfying Equation (3).
- Step 8: The PSO model applies the SPV rule to determine the panel sequence for each particle.
- Step 9: The simulation model runs for each particle and stores the total makespan.
- Step 10: The PSO updates the personal best value (p_{ld}) and position (x_{ld}) for each particle. If the current fitness value ($f^{\,k}$) is less than the personal best value (p_{ld}), then $p_{ld} = f^{\,k}$ and the personal best position, $x_{ld} = x_{id}$.
- Step 11: The PSO model updates the global best value (p_{gb}) and position (x_{gd}) by taking the minimum value of the personal best and the corresponding position value. Furthermore, the global best value, $p_{gb} = min\{p_{ld}\}$. If the fitness value for the current iteration ($f^{\,k}$) is less than the global best value (p_{gb}), then $p_{gb} = f^{\,k}$ and the global best position, $x_{gd} = x_{id}$.
- Step 12: If the number of iteration exceeds the maximum number of iteration, the PSO model stops the search; otherwise, it advances to step 4. Once the search stops, the optimum panel sequence is

defined as the resultant sorted global position values from smallest to largest, while the global fitness value is defined as the total production time.

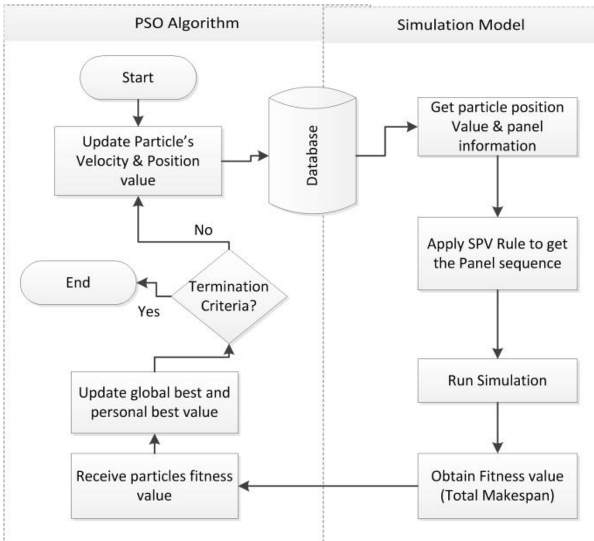

Figure 4. Integration between PSO algorithm and simulation model.

5 Results and discussion

Initially, the optimization model is run for different iteration and particle numbers for 50 multi-wall panels in order to find the optimal number of particles and iterations by comparing the model result with the runtime. The model is run in Intel® Core™ i7 CPU (3.20 GHz). Table 1 presents the model runtimes and makespan corresponding to the optimal sequences for different numbers of particles and iterations. The total production time of all panels (makespan) varies from 512~550 min. The result shows that 20 particles and 20 iterations can provide satisfactory results within a reasonable model runtime of 26 min, while increasing the numbers of particles and iterations to greater than 20 does not have a significant effect on the optimization results.

Table 1 Model runtimes and makespans for different particle and iteration numbers (50 multi-wall panels).

Particle number	Iteration number	Model runtime (hr:min)	Makespan for optimal sequence (min)
5	5	0:02	550
20	10	0:15	540
20	10	0:14	523
20	20	0:26	514
20	20	0:26	522
20	20	0:27	524
30	20	0:40	523
10	40	0:27	546
30	15	0:31	526
30	30	1:03	514
30	30	1:00	521
30	30	1:01	512
30	50	1:42	522

The optimization model is run for different production dates in the LBS plant for 20 particles and 20 iterations. The simulation model, meanwhile, is run 10 times for the actual and optimal sequences, respectively, to provide the mean makespan. The result is summarized in Table 2. The model provides the mean makespan for the actual sequence, the mean makespan time for the optimal sequence, along with the maximum and minimum production times. The productivity improvement is calculated by comparing the mean makespans of the actual and optimal sequences. The results show that the productivity can be improved up to 10% by implementing this optimization model.

Table 2 Optimization results.

No. of multi-wall panel produced	Mean Makespan-Actual sequence (min)	Min. Makespan-Optimal sequence (min)	Mean Makespan-Optimal sequence (min)	Max. Makespan-Optimal sequence (min)	Productivity Improvement
38	489	440	454	475	7%
66	691	665	681	705	1%
68	744	658	670	691	10%
57	628	551	573	593	9%
70	775	713	728	746	6%

Figure 5(a) shows the convergence chart of the PSO search for all particles (400 iterations) and Figure 5(b) shows the convergence chart of one particle (20 iterations). The optimum result was found after approximately 200 iterations of all particles. After this point the particles did not find any better solution.

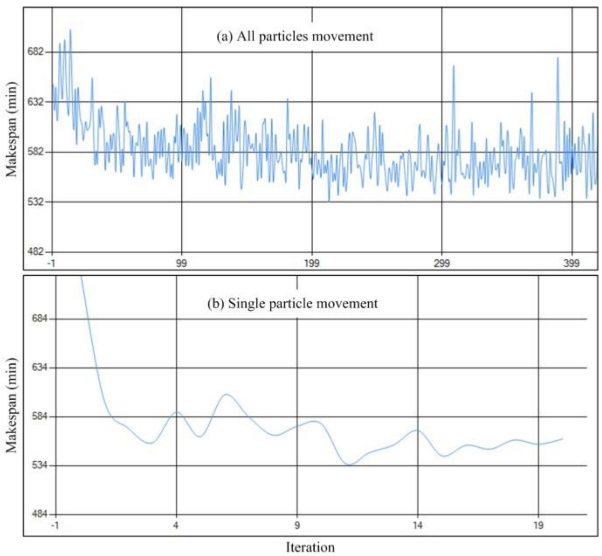

Figure 5. (a) Movement of all particles in the search area; (b) Single-particle movement in the search area

6 Conclusions

This paper present a research that integrated DES with a PSO algorithm in order to find the optimal panel sequence for a prefabricated wood-frame wall panel production line. The methodology is implemented on Landmark Group of Builders' prefabricated multi-wall panel production line. The simulation model of the production line is developed in Simphony.NET and integrated with a PSO algorithm using a central database containing multi-wall panel information. The optimization model is run for several actual production dates, and the optimization results are compared in order to measure the productivity improvement associated with the utilization of the model. The successful implementation of the proposed method in an actual production line demonstrates the practical usefulness of this model within the evolving panelized home building industry.

In future, the proposed methodology can be further improved by implementing other search algorithms such as genetic, ant colony, and bee algorithms, and the results can be compared to find the best algorithm for this type of sequencing problem. The search algorithm can also be modified based on the given requirements of a production team and the results can be observed using the simulation model.

Acknowledgements

The authors wish to acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC). The participation and support of our collaboration partner for this research, Landmark Group of Builders Ltd. is also appreciated.

References

[1] National Association of Home Builders. 2009. "Fast facts for panelized homes." Online: http://www.nahb.org/generic.aspx?genericContentID=10310, Accessed: 24/10/2013.

[2] Xie H., Shen W., Hao Q., Yu H., and Tang Y. Production simulation for prefab housing facilities. *3rd International Construction Specialty Conference*. Ottawa, ON, 2011.

[3] Baker K.R. and Trietsch D. Principal of Sequencing and Scheduling. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2009.

[4] Leu S. and Hwang S. GA-based resource-constrained flow-shop scheduling model for mixed precast production. Automation in Construction, 11: 439–452, 2002.

[5] Li W. D., Ong S. K., and Nee A. Y. C. Optimization of process plans using a constraint-based tabu search approach. International Journal of Production Research, 42(10): 1955-1985, 2004.

[6] Lu M., Lam H., Dai F.. Resource-constrained critical path analysis based on discrete event simulation and particle swarm optimization. Automation in Construction, 17(6): 670–681, 2008.

[7] Mahdavi I., Shirazi B., and Sahebjamnia N. Development of a simulation based optimization for controlling operation allocation and material handling equipment in FMS. International Journal of Production Research, 49(23): 6981-7005, 2011.

[8] Shewchuk J.P. and Guo C. Panel sequencing, and stack locating in residential construction: lean approach. Journal of Construction Engineering

and Management, 138 (9):1006-1016, 2012.

[9] Hammad A., Senghore O., Hastak M., and Syal M. Simulation Model for manufactured housing processes. Computing in Civil Engineering, pp 286-297, 2002.

[10] Garza-Reyes J.A., Oraifige I., Soriano-Meier H., Forrester P.L., Harmanto D. The development of a lean park homes production process using process flow and simulation methods. Journal of Manufacturing Technology Management, 23(2): 178-197, 2012.

[11] Dengiz B. and Alabas C. Simulation optimization using tabu search. *Proceedings of the 2000 Winter Simulation Conference*, pages 805-810, Orlando, FL, 2000.

[12] Rezg N., Xie X., and Mati Y. Joint optimization of preventive maintenance and inventory control in a production line using simulation. International Journal of Production Research, 42(10): 2029-2046, 2004.

[13] AbouRizk S. and Mohamed Y. Simphony - an integrated environment for construction simulation. *Proceedings of the 2000 Winter Simulation Conference*, pages 1907-1914, Orlando, FL, 2000.

[14] Kennedy J. and Eberhart R.C. Particle swarm optimization. *Proceeding of IEEE International Conferenceon Neural Networks*, Vol 4, pages 1942-1948, New York, USA, 1995.

[15] Tasgetiren M.F., Sevkli M., Liang Y., and Gencyilmaz G. Particle swarm pptimization algorithm for single machine total weighted tardiness problem. Proc. Congress on Evolutionary Computation, CEC. Vol 2: 1412-1419, 2004.