

Routing of Mobile Resources with PSO using Chaotic Randomness (Chaotic-PSO) for Unexpected Delivery Failures in Manufacturing

Alper ÖZPINAR¹, Emel Şeyma KÜÇÜKAŞCI²

¹*Istanbul Commerce University, Faculty of Engineering and Design, Department of Mechatronics Engineering, Istanbul, Turkey.*

²*Istanbul Commerce University, Faculty of Engineering and Design, Department of Industrial Engineering, Istanbul, Turkey.*

ABSTRACT

Dealing with short term and long term production planning and scheduling has already been solved with different optimization and artificial methods and approaches. Under normal manufacturing conditions supply and demand progress controlled and supported by decision support systems, ERP and MRP software packages aiming maximum utilization of resources and minimizing the stocks aiming for JIT. These software packages becoming even more intelligent and proactive based on the data in the database systems. However all these systems start with initial assumption of under normal conditions of flow time ordering to delivering. In case of unplanned stops, failures, malfunctions, shortages of unplanned inventory levels alter these initial conditions and progress to a diverging outcomes and consequences. This paper aiming a dynamic allocation and routing of mobile resources in a manufacturing plant by reallocating them by using a modified Particle Swarm Optimization using Chaotic Randomness.

Keywords: Particle Swarm Optimization, Chaotic Functions, Resource Planning and Routing.

I. INTRODUCTION

The balance between the supply and demand has been dramatically changed within the last centuries also changing with the customer requirements. Assuming the industrial revolution has been started within the 18th century where low quality of output and moral laxity on the workforce [1] combined with unplanned actions without the usage of improved machinery except some drills for construction and small lathes for general purpose usage [2] .

As a result of demands driving force on production evolves it in to new way of production namely mass production. This evolution replaces the functional form of production with one based on flow, and the systematic reduction of individual tasks, had all been pioneered mainly by American car manufacturers. Even the conception of a utilitarian vehicle priced for mass consumption had been widely discussed and implemented with the “curved dash” Oldsmobile by American car-makers before the launch of the Ford’s famous Model T which has been produced more than fifteen million cars based mostly on the same design

and production line for twenty years [3]. Since the assembly lines have been a significant development for managing operations – a mode that allows high-volume, low-cost, standardized production after the success of Ford’s Model T production in the assembly line. These benefits are often offset by drawbacks: perceptions of Ford style assembly lines consider them to be rigid and inflexible [4]. This may be the main reason of producing the same model for long years as single color which is black. However since the demand higher than the supply even with the mass production of same model, this factor has never been powerful enough for making a change in the way of production.

Ford’s achievement results in high production rates comparatively for those years in the assembly line production. After the 2nd World War, still the demand is higher than the supply but the consumers looking for more choices and quality, another car manufacturing company Toyota makes a huge improvement and reset the rules of the manufacturing game that organizes manufacturing and logistics [5]. The new system named “The Toyota Production

System (TPS)” which is still commonly used in most of the biggest manufacturing companies.

TPS or in other words the “Toyota Way” can be considered as early development of the nowadays key manufacturing recipe the “Lean production”. Lean production aiming to manufacture without generating waste and scraps also is targeting the value with less work. The origin of Waste in manufacturing which is “muda” in Japanese [6] has seven different types namely; waste from over production, waste of waiting time, transportation waste, inventory waste, processing waste, waste of motion, and waste from product defects.

Manufacturing bodies namely factories plan and run their production systems based on the two basic concepts according to TPS. First of all, the thing that corresponds to the first recognition of putting forth all efforts to attain low cost production is “reduction of cost through elimination of waste”. This involves making up a system that will thoroughly. Eliminate waste by assuming that anything other than the minimum amount of equipment, materials, parts, and workers (working time) which are absolutely essential to production are merely surplus that only raises the cost. In order to obtain eliminating the wastes result in another popular way of production named as “Just in time-JIT” production. In JIT in order to avoid such problems as inventory unbalance and surplus equipment and workers, companies recognizes the necessity of schemes adjustable to conform with changes due to troubles and demand fluctuations. For this purpose, companies put their efforts in development of a production system which is able to shorten the lead time from the entry of materials to the completion of vehicle. The just-in-time production is a method where by the production lead time is greatly shortened by maintaining the conformity to changes by having “all processes produce the necessary parts at the necessary time and have on hand only the minimum stock necessary to hold the processes together”. [6-9] Minimum stock or stockless working principle requires a perfect material flow on the value chain model from supplier to the assembly shop. Even with the JIT operation conditions the material flow on the value chain model made as bulk or grouped transfers due to transportation costs. Most of the factories have main warehouses where the supplier delivers the products the factory area based on the long term weekly or monthly assumptions of production and Kanban systems pull the instant demand for the production. From the days of producing the same car type for decades nowadays manufacturing companies producing the products

according to the exact needs of the customer by means of colour, specs, extras, motor type, seat covers, electronic controls even the fuel octane filled into the tank.

An FMS (Flexible Manufacturing System) in a JIT (Just-In-Time) production system to be considered here consists various numbers of workstations labelled, dispatching stations and AGV (Automated Guided Vehicle) and tow trucks namely mobile transfers. AGV’s and tow trucks used for transportation between the different locations within the factory area as single or multiple loads. [10-12].

II. MATERIAL AND METHOD

2.1. Problem Overview

According to the original problem under normal manufacturing planning and production conditions mobile resources following the deterministic routes and planned operations, stock delivery and distribution cycles supported by the MRP and ERP systems. Since most of the logistics and transportation still operated by blue collar workers and not fully automated there is a possibility of stochastic and unplanned needs to fulfill Kanban operations such as misinformation, wrong parts delivery, barcode problems, jundate-ticket selection problems, wrong material flow, human error, database error, communication errors and expert system setting and definition errors. Most of these errors due to the rapid development and changes in the standard operating procedures within the factories since the needs for starting the manufacturing operation is most of the time called last minute changes in the system. [5] The problem is originated from the work of the authors based on the AGV routing problem of handling the unplanned equipment transfer. [8] The formulation of the problem is also altered to be formulated as a network optimization problem.

In order to simulate the problem case a general approach for a typical manufacturing factory layout assumed. The factory area is considered as a Euclidean space in a 90x110 rectangular area. Three main warehouses comprising ten shelves provide the main supply in a weekly time table. As a result there are thirty locations and these locations are distant to the manufacturing area. More closer to the assembly lines, ten assembly warehouses are located. The workers fill the products in these locations based on MRP results in an hourly time table. In order to decrease the work in process inventory, the assembly warehouse capacity is

finite and limited. Since there may be different production specifications of similar products manufactured in distinct lines, it is reasonable to consider ten assembly lines located closer to the assembly warehouses. The display of the factory layout is shown in Figure 1.

There are three basic locations within an ideal manufacturing based on Toyota Production Practices. Supplier delivers the materials based on the long term planning and the sales to the MW Main Warehouse. During the midterm planning week or day materials are taken from the main warehouse location to the assembly warehouse destination (MW-AW job). Depending on the tag times and Kanban system, materials are taken from assembly warehouse location to the assembly line destination where they used in the manufacturing line (AW-AL job). These transfer operations mostly done by AGV's or tow truck vehicles. For this papers scenario it has been assumed that there are two types of AGVs such that type 1 AGV and type 2 AGV controlled by a computer or system center. Both types of AGVs identical like the size, capacity, speed but only difference between them emanates from the assigned distinct travelling areas. When there is a missing part in an assembly line, that part must be picked up from a main warehouse then transfer of the part to the assembly warehouse is a MW-AW job. Latter two jobs are done with a type 1 AGV. At the end, the part is delivered to the demanding assembly line through an AW-AL job by a type 2 AGV.

The problem can be constructed as a minimum cost flow (MCF) problem taking the inspiration from a paper which formulates the MCF problem for AGV routing in container terminals [13]. In Figure 2 $G=(N,A)$, each initial AGV location can be considered as a source node. In addition, each job is represented with a couple of nodes and the corresponding arc between them. This representation permits only one visit to each main warehouse location, and also enables an AGV to visit multiple main warehouses. The arcs represent the different mobile resources jobs. All arc capacities must be limited by 1 for planning. Finally, a sink node has used for the satisfaction of the demands of all jobs. The arc costs are determined using the travelling times of the AGVs based on the travelling distances. The main objective of the problem is to minimize the total travelling distance of the AGVs calculated after all the requested jobs are performed. The described directed graph is illustrated in Figure 2 for two jobs

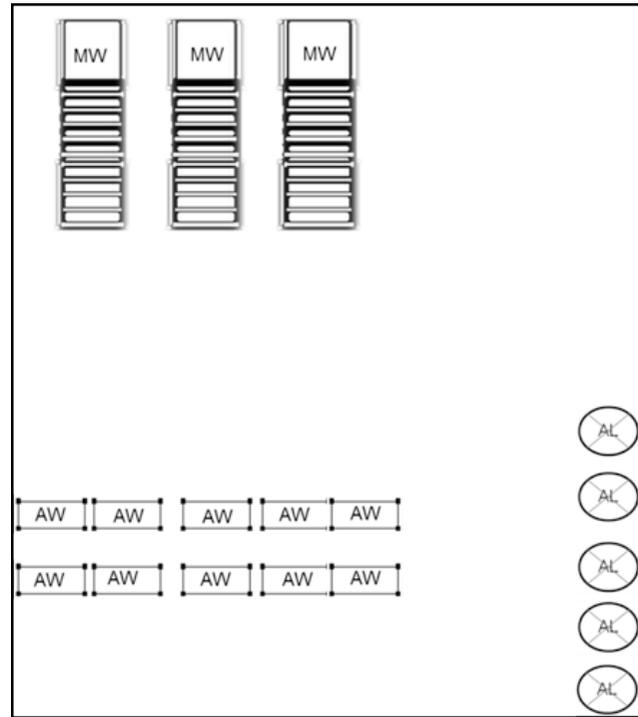


Figure 1. Schema of the Factory Layout for Simulation

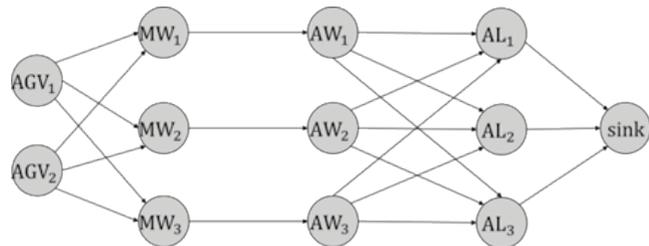


Figure 2. A Network Example for the MCF Formulation of AGV Routing Problem

2.2. Mathematical Model

The description and the nomenclature of the model parameters and the variables are as follows:

- AGV : set of type 1 AGVs,
- I : set of initial AGV location nodes,
- M : set of main warehouse nodes,
- A : set of assembly warehouse nodes,
- A' : set of assembly line nodes,
- s : the sink node,
- $n = |AGV|$,
- $A_1 = \{(i, j) : i \in I, j \in M\}$,
- $A_2 = \{(i, j) : i \in M, j \in A\}$,
- $A_3 = \{(i, j) : i \in A, j \in A'\}$,
- $A_4 = \{(i, s) : i \in A'\}$,

$$c_{ij} = \begin{cases} d_{IM}, & \text{if } (i, j) \in A_1 \\ d_{MA}, & \text{if } (i, j) \in A_2 \\ d_{AA}, & \text{if } (i, j) \in A_3, \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{if arc } (i, j) \text{ is traversed,} \\ 0 & \text{otherwise.} \end{cases}$$

The complete model is as follows:

$$\text{Min} \left\{ \sum_{(i,j) \in A} c_{ij} \cdot x_{ij} \right\} \tag{1}$$

s.t.

$$\sum_{j \in N: (i,j) \in A_1} x_{ij} = 1, \quad \forall i \in AGV \tag{2}$$

$$\sum_{i \in N: (i,j) \in A_4} x_{is} = n, \tag{3}$$

$$\sum_{i \in N: (i,j) \in A} x_{ij} - \sum_{i \in N: (j,i) \in A'} x_{ji} = 0, \forall j \in A_1 \cup A_2 \cup A_3 \tag{4}$$

$$0 \leq l_{ij} \leq x_{ij} \leq u_{ij}, \forall (i, j) \in A$$

where

$$l_{ij} = 0, u_{ij} = 1 \text{ if } (i, j) \in A_1 \cup A_3 \text{ and } l_{ij} = 1, u_{ij} = 1 \text{ if } (i, j) \in A_2 \cup A_4.$$

The objective function (1) minimizes the total traversed distance. The constraint (2) satisfies that there is only one arc outgoing arc exists at type 1 AGV nodes. The constraint “(3)” satisfies the condition that the sink node s has only n much incoming arcs and no outgoing arcs. Finally (4) is the flow balance constraint and (5) corresponds to the flow capacity constraint.

The initial coordinates of the AGVs are generated from a uniform distribution between 0 and 90. Since the main objective is to obtain an efficient schedule, the distances between all locations are calculated as a Euclidean distance and we assumed that there is no occurrence of collusions, jamming or waiting times. All the methods start with the same initial AGV locations. Additionally, same target location parameters are also given as an input to all methods. Number of the emergency type 1 AGVs is equal to the

number of the jobs that are targeted from the initial AGV locations to the main warehouses, namely MW-AW jobs. Similarly, number of the emergency type 2 AGVs is equal to the number of AW-AL jobs.

Due to the sequencing constraints, the described problem is more complex than the multi travelling salesperson problem (MTSP) which is NP-complete [14]. Hence, being at least as hard as the MTSP, AGV routing problem is NP-hard. In order to find a grateful solution in a reasonable time, metaheuristic algorithms and/or their variations are introduced [15-18]. Chaotic-PSO is a variant of the previous PSO approach proposed for solving the unplanned AGV routing problem. [8] This is the main contribution of this paper which clearly exhibits the difference with PSO and the Chaotic-PSO approaches.

In this section special implementations of the PSO and Chaotic-PSO algorithms suited to the AGV routing problem are described. The famous PSO approach is presented first. Then, the Chaotic-PSO algorithm is discoursed together with the various chaotic functions.

2.3. PSO for Mobile Resource Routing

The modified PSO algorithm is a discrete solution for the dynamic routing of the mobile resources like emergency AGVs which can be changed during the operation according to the particle with the best-found travelling distance in whole preceding iterations and the particle with the best-found travelling distance in current the iteration. A particle represents a solution that includes the routes of the AGVs satisfying the all requested stock delivery jobs. Positions of the particles in iteration k , x^k form the population which is updated at each iteration using the velocity vector. Velocity vector at iteration k , v^k is calculated using the inertia weight w , learning factors c_1 and c_2 , and the random numbers λ_1 and λ_2 . The fitness of each particle is determined by considering the total traversed distances. The inertia weight in the k^{th} iteration is calculated by the formula

$$\omega^k = \omega_{mzx} - (k / k_{mzx}) (\omega_{mzx} - \omega_{min}). \tag{5}$$

where k_{mzx} is the maximum number of iterations, is the current iteration number and ω_{min} and ω_{mzx} are the minimum and the maximum values for w , respectively.

Afterwards, v^k is calculated using the equation

$$v^{k+1} = \omega^k \cdot v^k + c_1 \cdot \lambda_1 \cdot (p_{Best} - x^k) + c_2 \cdot \lambda_2 \cdot (g_{Best} - x^k) \tag{6}$$

where p_{Best} is the best-found travelling distance in whole preceding iterations and g_{Best} is the best-found travelling distance in current the iteration.

Since the problem has a discrete solution, addition operation of the velocity vector to the particle positions not only results with rational positions, but also results with deteriorating particle positions. Therefore, the update of the current position of the particles in the population is done by a swap operator. Several swap operators are considered in some PSO approaches to the TSP problem [15;19;20]. The proposed swap operator is such that the particles with the minimum and the maximum velocity values are swapped first. Together with this operation, the particles with the second minimum velocity and the second maximum velocity are also swapped. Since the swapping operations do not harm the completeness of the particles, the feasibility is sustained after the swap operations. Hence, the positions of the particles in the population are updated according to the following equation where the \oplus sign is the swap operator.

$$x^{k+1} = x^k \oplus v^{k+1} \quad (7)$$

Finally, the general PSO algorithm used is as follows:

For each particle

Initialize particle

End

While iteration number < maximum number of iterations

Do

For each particle

Calculate fitness value

If the fitness value is better than the best fitness value (p_{Best})

in history

set current value as the new p_{Best}

End

Choose the particle with the best fitness value of all the particles as the g_{Best}

For each particle

Calculate particle velocity according to equations (5) and (6)

Update particle position according to equation (7)

End

End

2.3. PSO with Chaotic Randomness (Chaotic-PSO) for Mobile Resource Routing

To cope with the random calculation of λ_1 and λ_2 values at each iteration, selected chaotic functions are included. Moreover, combined with chaos optimization algorithm, the particles can go through all places in the solution space without repetition under a deterministic iteration formula. [15] The diversification of the search and also the time complexity is improved by embedding the chaotic functions instead of the random variables.

One dimensional and two dimensional chaotic maps are applied to solve test task scheduling problem (TTSP) to replace random sequences. [18] One-dimensional chaotic maps with promising results are selected as chaotic functions to improve the PSO algorithm. Hence, the λ_1 and λ_2 variables in (6) are updated using the chaotic maps which are shown in Table 1. Besides, the positions of the particles are updated using (7) where the \oplus sign is the swap operator defined previously.

Table 1. List of Chaotic Maps for Randomness

| Chaotic map | Formula | Range |
|----------------|----------------------------------------------------------------------------------------|-------|
| Circle map | $x^{t+1} = \{x^t + b - (a/2\pi) \sin(2\pi x^t)\} \text{ mod}(1)$, $a = 0.5$ $b = 0.2$ | (0,1) |
| Cubic map | $x^{t+1} = \rho x^t (1 - (x^t)^2)$, $\rho = 2.59$ | (0,1) |
| Logistic map | $x^{t+1} = a x^t (1 - x^t)$, $a = 4$ | (0,1) |
| Sinusoidal map | $x^{t+1} = \sin(\pi x^t)$ | (0,1) |

2.4. Experimental Settings and Simulation Results

General experimental setting is shown in Table 2. Since the scheduling is applied only for the emergencies, there are less

number of jobs than a daily or weekly delivery schedule. At most 20 jobs are considered at a single time interval in order to see the situation of the algorithms.

Table 2. Parameters for the Simulation Environment

| Parameter | PSO | Chaotic-PSO |
|-------------------------|--------------|-------------|
| c_1, c_2 | 0.5, 0.5 | 0.5, 0.5 |
| λ_1, λ_2 | $\in U(0,1)$ | Chaotic |
| Population size | 150 | 150 |
| w_{min}, w_{max} | 0.5, 0.95 | 0.5, 0.95 |
| Maximum # of iterations | 300 | 300 |

Testing the suitability is done for 16 problem instances based on job numbers. The instances are incremental such that each instance is a subset of the following large-sized instance in Table 3. The followings are the main observations after reaching the results in Table 3. When there are 5 jobs, both PSO and Chaotic-PSO succeeded to find the solution with same objective value. However, when we move forward to the larger instances, the solutions found by each method differ from each other which may be due to the increasing complexity of the problem. For all instances, Chaotic-PSO with logistic map gives the best scores. Second best results are obtained by the circle map implementation of Chaotic-PSO for several instances. When we check the score averages, third best approach is the cubic map integrated Chaotic-PSO. Between all Chaotic-PSO approaches, the worst results are obtained by using the sinusoidal map as a chaotic function. The averages tell us that the classical PSO algorithm performs not well when the competitor is circle map, logistic map or cubic map integrated Chaotic-PSO. Finally, it is worth-emphasizing that the inclusion of chaotic position updates prevents the early convergence of the PSO algorithm.

The objective function (1) minimizes the total traversed distance. The constraint (2) satisfies that there is only one arc outgoing arc exists at type 1 AGV nodes. The constraint “(3)” satisfies the condition that the sink node s has only n much incoming arcs and no outgoing arcs. Finally (4) is the flow balance constraint and (5) corresponds to the flow capacity constraint.

The initial coordinates of the AGVs are generated from a uniform distribution between 0 and 90. Since the main objective is to obtain an efficient schedule, the distances between all locations are calculated as a Euclidean distance and we assumed that there is no occurrence of collisions, jamming or waiting times. All the methods start with the

same initial AGV locations. Additionally, same target location parameters are also given as an input to all methods. Number of the emergency type 1 AGVs is equal to the number of the jobs that are targeted from the initial AGV locations to the main warehouses, namely MW-AW jobs. Similarly, number of the emergency type 2 AGVs is equal to the number of AW-AL jobs.

Due to the sequencing constraints, the described problem is more complex than the multi travelling salesperson problem (MTSP) which is NP-complete [14]. Hence, being at least as hard as the MTSP, AGV routing problem is NP-hard. In order to find a grateful solution in a reasonable time, metaheuristic algorithms and/or their variations are introduced [15-18]. Chaotic-PSO is a variant of the previous PSO approach proposed for solving the unplanned AGV routing problem. [8] This is the main contribution of this paper which clearly exhibits the difference with PSO and the Chaotic-PSO approaches.

In this section special implementations of the PSO and Chaotic-PSO algorithms suited to the AGV routing problem are described. The famous PSO approach is presented first. Then, the Chaotic-PSO algorithm is discoursed together with the various chaotic functions.

III. RESULTS AND DISCUSSION

Testing the suitability is done for 16 problem instances based on job numbers. The instances are incremental such that each instance is a subset of the following large-sized instance in Table 3. The followings are the main observations after reaching the results in Table 3. When there are 5 jobs, both PSO and Chaotic-PSO succeeded to find the solution with same objective value. However, when we move forward to the larger instances, the solutions found by each method differ from each other which may be due to the increasing complexity of the problem. For all instances, Chaotic-PSO with logistic map gives the best scores. Second best results are obtained by the circle map implementation of Chaotic-PSO for several instances. When we check the score averages, third best approach is the cubic map integrated Chaotic-PSO. Between all Chaotic-PSO approaches, the worst results are obtained by using the sinusoidal map as a chaotic function. The averages tell us that the classical PSO algorithm performs not well when the competitor is circle map, logistic map or cubic map integrated Chaotic-PSO. Finally, it is worth-emphasizing that the inclusion of chaotic position updates prevents the early convergence of the PSO algorithm.

Table 3. P Total distance travelled by mobile resources for jobs

| #JOBS | PSO | CHAOTIC-PSO (CIRCLE MAP) | CHAOTIC-PSO (CUBIC MAP) | CHAOTIC-PSO (LOGISTIC-MAP) | CHAOTIC-PSO (SINUSOIDAL MAP) |
|-------|---------------|-----------------------------|----------------------------|-------------------------------|------------------------------------|
| 5 | 2438.2 | 2438.2 | 2438.2 | 2438.2 | 2438.2 |
| 6 | 2902.9 | 2904.3 | 2902.9 | 2899.2 | 2907.6 |
| 7 | 3198.1 | 3174.6 | 3187.7 | 3164.1 | 3198.1 |
| 8 | 3585.7 | 3584.1 | 3585.7 | 3581.9 | 3588.2 |
| 9 | 4042.8 | 4041.3 | 4041.3 | 4030.3 | 4045.4 |
| 10 | 4577.2 | 4530.1 | 4544.7 | 4513.6 | 4553.0 |
| 11 | 4956.0 | 4950.6 | 4951.2 | 4920.9 | 4957.6 |
| 12 | 5485.8 | 5437.8 | 5441.8 | 5431.7 | 5475.3 |
| 13 | 6085.8 | 6045.5 | 6034.6 | 6028.7 | 6048.5 |
| 14 | 6408.0 | 6401.4 | 6400.8 | 6376.0 | 6410.1 |
| 15 | 6866.1 | 6841.1 | 6848.0 | 6834.2 | 6861.5 |
| 16 | 7073.3 | 7073.4 | 7075.1 | 7061.4 | 7078.0 |
| 17 | 7600.1 | 7553.4 | 7585.8 | 7530.0 | 7590.1 |
| 18 | 7851.7 | 7788.3 | 7839.2 | 7776.7 | 7879.2 |
| 19 | 8223.8 | 8213.9 | 8223.2 | 8149.9 | 8291.0 |
| 20 | 8722.2 | 8693.3 | 8711.0 | 8639.9 | 8778.7 |
| AVG. | 5626.1 | 5604.4 | 5613.2 | 5586.0 | 5631.3 |

IV. CONCLUSION

Recently, use of particle swarm optimization for different scheduling and resource allocation problems has been broadly used and offered. In this paper, both general particle swarm optimization and a modified particle swarm optimization where the true random generator of the classical approach has been modified with a chaotic random number generation applied to mobile resource routing problem. Proposed algorithm and approach applied to an unexpected planning request generated within a JIT-Kanban operating manufacturing factory. Visual Studio and C# has been used for developing the simulation software.

According to the simulation results, Chaotic-PSO with a logistic-map function for randomness gives the optimum results by means of less distance travelled during the emergency operation of the mobile resources.

The future work for the researchers will be based on the variation of unexpected jobs and delivery problems by using a chaotic problem generator and a hybrid approach of C-PSO and GA and the performance measurements of the problem

REFERENCES

- [1] Behagg, C. (1998). Mass Production without the Factory: Craft Producers, Guns and Small Firm Innovation, 1790-1815, *Business History*, 40, 1-15.
- [2] Herer, Y. T., Shalom, L. (2000). The Kanban assignment problem "A non-integral approach", *European Journal of Operational Research*, 120, 260-276.
- [3] Klei, C. M., Kim, J. (1996). AGV dispatching, *International Journal of Production Research*, 34, 95-110
- [4] Lage Junior, M. & Godinho Filho, M. (2010). Variations of the kanban system: Literature review and classification, *International Journal of Production Economics*, 125, 13-21.
- [5] Lu, H, Yin, L, Wang, X, Zhang, M, & Mao, K. (2014). Chaotic Multiobjective Evolutionary Algorithm Based on Decomposition for Test Task Scheduling Problem, *Mathematical Problems in Engineering*.
- [6] Maughan, F. G. & Lewis, H. J. (2000). AGV controlled FMS. *International Journal of Production Research*, 38, 4445-4453.
- [7] Musson, A. E. (1975). Joseph Whitworth and the Growth of Mass-Production Engineering, *Business History*, 17, 109-149.
- [8] Ozpinar A., Kucukasci E. (2014). Particle Swarm Optimization

- for Unplanned AGV Routing in Kanban Systems, *15th EU/ME Workshop, "Metaheuristics and Engineering"*.
- [9] Rahman, N. A. A, Sharif, S. M., Esa, M.M. (2013). Lean Manufacturing Case Study with Kanban System Implementation, *Procedia Economics and Finance*, 7, 174-180.
- [10] Rashidi, H. & Tsang, E. P. K. (2011). A complete and an incomplete algorithm for automated guided vehicle scheduling in container terminals, *Computers. & Mathematics with Applications*, 61, 630-641.
- [11] Samanlioglu, F, Ferrell, W. G, & Kurz, M. E. (2012). An interactive memetic algorithm for production and manufacturing problems modelled as a multi-objective travelling salesman problem, *International Journal of Production Research*, 50, 5671-5682.
- [12] Shi, X. H, Liang, Y. C, Lee, H. P, Lu, C, & Wang, Q. X. (2007). Particle swarm optimization-based algorithms for TSP and generalized TSP, *Information Processing Letters*, 103, 169-176.
- [13] Somhom, S., Modares, A., Enkawa, T. (1999). Competition-based neural network for the multiple travelling salesmen problem with minmax objective, *Computers & Operations Research*, 26, 395-407.
- [14] Sugimori, Y., Kusunoki, K., Cho, F., Uchikawa, S. (1977). Toyota production system and Kanban system Materialization of just-in-time and respect-for-human system, *International Journal of Production Research*, 15, 553-564.
- [15] Wallace, A. (2001). Application of AI to AGV control agent control of AGVs. *International Journal of Production Research*, 39, 709-726.
- [16] Wang, K. P, Huang, L, Zhou, C. G, & Pang, W. (2003). Particle swarm optimization for traveling salesman problem, *Machine Learning and Cybernetics*, 3, 583-1585.
- [17] Wilson, J. M. (2013). Henry Ford vs. assembly line balancing, *International Journal of Production Research*, 52, 757-765.
- [18] Wilson, J. M. & McKinlay, A. (2010). Rethinking the assembly line: Organisation, performance and productivity in Ford Motor Company, c. 1908-27, *Business History*, 52, 760-778.
- [19] Yuan, Z, Yang, L, Wu, Y, Liao, L, & Li, G. (2007). Chaotic particle swarm optimization algorithm for traveling salesman problem, *IEEE International Conference on Automation and Logistics*, 1121-1124.
- [20] Zhang, Q, Manier, H, & Manier, M. A. (2012). A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded processing times, *Computers & Operations Research*, 39, 1713-1723.