ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)

Volume 11 Issue 8 August 2024

Applied Modified Particle Swarm Optimization to Dynamic Lot Sizing with Customer Orders Problem Considering

[1] Sheng-Yuan, Hsu

[1] Department of Industrial Engineering and Service Management, Chienkuo Technology University, Taiwan
Corresponding Author Email: [1] jackhsu@ctu.edu.tw

Abstract— ***On time delivery is a very important issue for customers in supply chain management. The customer's order includes more than one item at all times. How to finish the order on time, so that all items in the same order will be ready for delivery, is an important task. Therefore, this paper considers the dynamic demand lot-sizing problem (DLSP) and the customer-ordering problem (COP) together, namely DLSCOP, dynamic lot sizing problem with customer order considering. DLSP focuses on the deterministic time-varying batch ordering lot-sizing problem with backorders. The COP consists of a set of items that must be shipped as one batch at the same time.***

This work applies a modified particle swarm optimization (mPSO) to solve the problem. Two popular algorithms, Silver-Meal (SM) algorithm and Wagner-Whitin (WW) algorithm, for benchmarking are modified and two heuristics MSM, MWW are developed for solving DLSCOP. The genetic algorithm (GA) will be included in the simulation experiment for comparing. The simulation test considers 128 scenarios and 100 repetitions. In the statistical analysis, the mPSO performance is better than GA, MSM and MWW. The decision based on MPSO saves more than 10-50% cost, especially in those scenarios with long term, multiple items, and high expense rate (ordering cost and holding cost).

Keywords: dynamic lot sizing, customer order problem, particle swarm optimization.

I. INTRODUCTION

In supply chain management, customer order should be delivered on time for the customer to proceed their processes or delivery to their customer. Customer order always orders more than one items and those items should be delivery at the same ship for fulfilling customer. This kind of problem namely COP, customer order problem, is very common in supply chain management. Customers issue an order based on their own order. The order will order more than one items for fulfilling their customer's demand. All items in the same order require same time delivery, and the order is not ready for delivery if any item in the same order is not ready for shipment. This type of problem is a typical COP, like as figure 1. Furthermore, lot sizing problem is a traditional problem for inventory and production management. DLSP is the dynamic lot sizing problem. DLSP focuses on the deterministic time-varying batch ordering lot-sizing problem with backorders. The objective is to determine the optimum ordering plan, i.e. minimizing the total cost, to satisfy a set of known demands over a specific planning horizon.

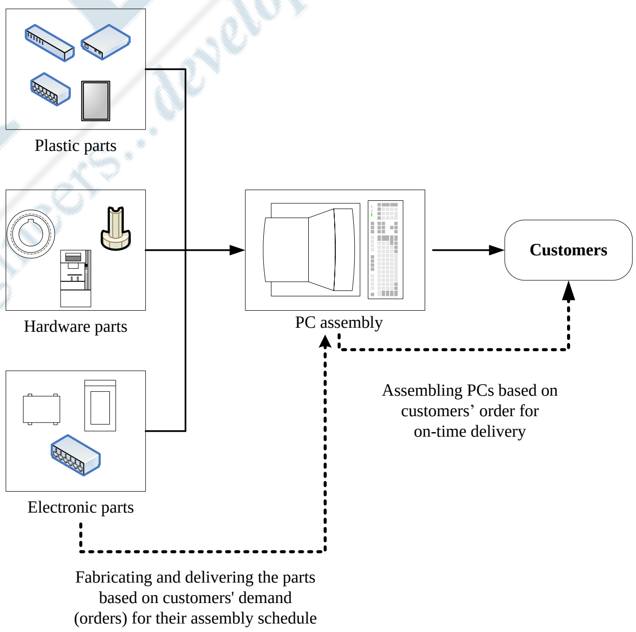

Figure 1. Customer ordering problem in the supply chain of personal computer

This research investigates the applicability of particle swarm optimization (PSO) on dynamic lot sizing problem with customer order considering, namely DLSCOP. DLSCOP focus on two important problems in supply chain management, that is DLSP and COP. DLSP is the dynamic lot sizing problem. COP represents for the customer-ordering problem. In a traditional dynamic lot-sizing problem, the seller makes the purchasing decision based on the customers'

ISSN (Online) 2456 -1304
International Journal of Science, Engineering and Management (IJSEM)
Volume 11 Issue 8 August 2024

order. In practice, the customer order includes more than one item in the same order. All items in the same order require delivery in the same shipment for proceeding to the next fabrication step. Hence, the dynamic lot sizing decision should consider the COP.

Figure 2 is a typical example of DLSCOP. There are some orders in the same period. All the items' total demand at each period consists of the demand of some orders. In period 1, the total demand of item 1 is 10, consisting with the demand of order 1, 2, 3 for item 1. The items in the same order should be delivered at the same ship. DLSP tries to find a suitable purchasing decision for minimizing total inventory cost. When the DLSP considers the customer order problem, the purchasing decision will be more complex. This work develops a linear programming model to describe the DLSCOP and developed a modified particle swarm optimization, mPSO, for sloving the problem. Genetic algorithms (GA) and some heuristics methods are considered for comparing.

Section 2 discusses the relative researches of COP and DLSP. Section 3 presents the problem formulation of DLSCOP and the LP model with a brief analysis. Section 4 discusses the PSO approach for solving DLSCOP and section 5 presents the computation experiment. Sections 6 discuss the analytical result and have some discussion.

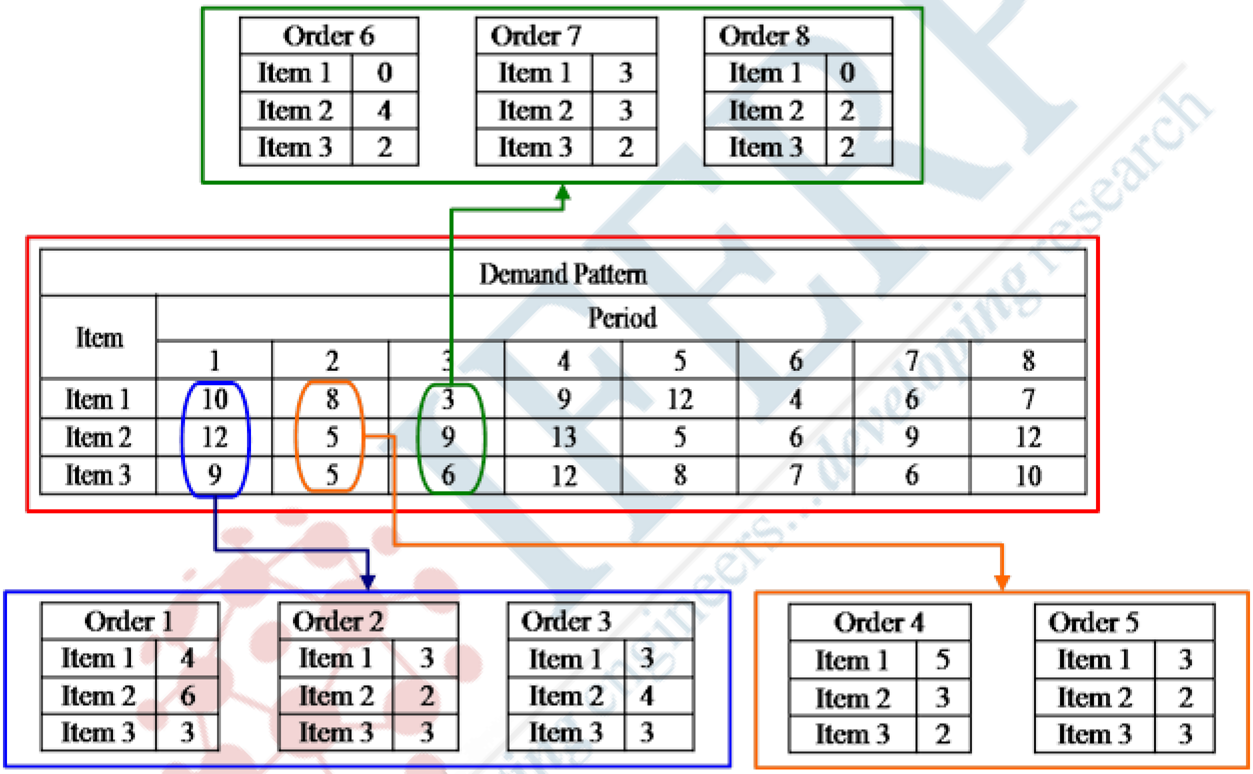

Figure 2. the example of dynamic lot sizing problem with customer order considering

II. LITERATURE REVIEW

A. DLSP

Lot sizing models determine the optimal timing and level of purchasing or production. One end of the spectrum includes the continuous time scale, constant demand, and infinite time horizon lot sizing problems. In this category we find the famous economic order quantity model (EOQ) and the economic lot scheduling problem (ELSP). The other end of this spectrum includes the discrete time scale, dynamic demand, and finite time horizon lot sizing models (Jans and Degraeve, 2007). Studies typically refer to this type of planning as dynamic lot sizing and it is the main subject of this paper. A number of studies have introduced the lot-sizing problem. De Bodt et al. (1984) and Bahl et al. (1987) presented earlier reviews. Jans and Degraeve (2007) provided a wider survey of meta-heuristic applications in dynamic lot sizing. Robinson et al. (2009) updated a 1988 review of the coordinated lot-sizing problem and complemented recent reviews on the single-item lot-sizing problem and the capacitated lot-sizing problem, providing a state-of-the-art review of the research and future research projections. Adetunji and Yadavalli (2012) discussed the reducing of the batch sizes of products in production network through the utilization of the idle time. Kim and Lee (2013) considered a dynamic inbound ordering and shipment scheduling problem for lot sizing problem. These studies have presented a complete framework for the lot-sizing problem and most have discussed the model and algorithms. Different types of lot-sizing problems result from certain parameters, such as static/dynamic demand, incapacitated/capacitated, lot sizing, backorder, setup cost/purchasing cost, etc. However, none of the researches in our review focused on the lot-sizing problem, considering the scenario of the customer order problem.

One of the most commonly used suboptimal algorithms for the methodology for solving DLSP, is the Silver-Meal (Silver and Peterson, 1985). The original SM algorithm finds the number of periods for minimizing the total inventory costs per period and then orders the exact quantity to cover the demand for those periods. Wagner and Whitin (1958) developed a dynamic programming algorithm to obtain the optimum solution to a simpler version of the lot-sizing problem, and developed several extensions of the original WW algorithm over time. However, in the review of Jans and Degraeve (2007), and Robinson et al. (2009), the lot-sizing problems are very complex to find the optimal solution. They presented many meta-heuristics for dynamic lot sizing, such as tabu search (TS), simulated annealing (SA), and genetic algorithms (GA).

Several authors have applied GAs to various versions of the lot-sizing problem. Ozdamar and Birbil (1998) utilized GA, simulated annealing (SA), and tabu search to solve the capacitated lot sizing problem on parallel machines. Hung and Chien (2000) also utilized GA, SA, and tabu search to solve the multi-class multi-level capacitated lot-sizing problem. Khouja et al. (1998) investigated using GAs to solve the economic lot size-scheduling problem using the basic period approach. Prasad and Chetty (2001) applied GAs to multi-level lot sizing and observed under a rolling horizon environment. The performance of GAs is superior to popular heuristics.

Staggemeier et al. (2002) presented a hybrid GA to solve a lot sizing and scheduling problem by minimizing inventory and backlog costs of multiple products on parallel machines with sequence-dependent set-up times. Basnet and Leung (2002) presented a multi-period inventory lot-sizing scenario, with multiple products and multiple suppliers. Sarker and Newton (2002) developed a GA code with three different penalty functions to determine optimal batch sizes for products and a purchasing policy for associated raw materials. They compared GA results to optimal solutions, and the GA with a static penalty function obtained the global optimum 100% of the time. Moon et al. (2002) developed a hybrid GA to address the lot-scheduling problem with time-varying lot sizes. A numerical experiment showed that the hybrid GA performed better than other heuristic methods. Shittu (2003) used GAs to address the lot-sizing problem with batch sizing and compared their performance to that of the SM.

B. COP

The customer order problem (COP) considered orders consisting with a set of items that must be shipped as one batch before the due date. It's a specific problem in the supply chain management. COP will be very important for the cooperation with those supply chain's members. The purchase order specifies the composition of an order. The company may not ship the order until all the items in the order are completed and ready for delivery. Figure 1 is an example of the personal computer assembly.

In the review of Hsu and Liu (2009), only a few studies focused on the customer order-scheduling problem. Most of the studies focused on the single machine or parallel machine system and tried to find an optimal solution. Julien and Magazine (1990) assumed a job-dependent setup time for two different types of jobs. They assumed a dynamic programming (DP) algorithm for the single machine problem with only two types of jobs and a fixed batch processing order. Coffman et al. (1989) considered the same problem as that of Julien and Magazine (1990), assuming a non-fixed batch processing order. In addition, Baker (1988), Gupta et al. (1997), and Gerodimos et al. (2000) also focused on the single machine case. Yang (2005) introduced a relatively new class of the COS case for parallel machines. He proposed an optimal solution procedure for each of several problems with different types of objectives, job restrictions, and machine environments. Yang and Posner (2005) considered the COS case in four jobs dispatched in batches. They presented a heuristic for the problem, and found a tight worst case bound on the relative error. Ahmadi et al. (2005) and Wang and Cheng (2007) considered the COS case for m dedicated facilities and n orders, in which each job only needs one operation at a dedicated facility. They showed that the problem is unary NP-hard, and proposed a heuristic method to minimize the total weighted order completion time and analyzed a worst-case scenario. Daganzo (1989) and Peterkofsky and Daganzo (1990) focused on the crane scheduling problem, which is simply another kind of COS problem with a parallel machine. In their discussions, jobs consist of independent, single stage, preemptible tasks. The objective function is minimizing weighted tardiness. Daganzo (1989) applied the weighted shortest processing time first rule to obtain optimal solutions for some special cases. Peterkofsky and Daganzo (1990) developed a branch-and-bound solution procedure. However, they did not provide a theoretical basis for their method.

Blocher et al. (1998) dealt with the COS problem in a general job shop consisting of six machines. Theirs is the first simulation study using the customer order environment. They specifically compared the dispatching rules from past job-based studies to rules adapted to include order characteristics. They divided performance measures into two parts: measures involving order flow time and measures involving due dates. The order flow-time measure is similar to the common job flow-time measure, except for the fact that the flow time is based on order parameters (groups of jobs). The due date measures are based on average tardiness and proportion tardy. Of the sixteen dispatching rules tested, four simple rules dominate all others. Those rules consider order characteristics, namely order-based rules, and perform better

ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)

Volume 11 Issue 8 August 2024

than their job-based counterparts perform.

Hsu and Liu (2009) discussed reducing the stock level of finished goods and improving delivery efficiency for the COP in the job shop. The main topic focused on how to control the finished time of all jobs in the same order in a normal job shop.

Given the discussion above, DLSP is one of the key problems in production planning, inventory management, and supply chain management. The purchasing decision is a very important issue, especially in cooperation with supply chain members. Studies need to consider how to make a quality decision for improving whole supply chain performance for competition advantage. COP is also another popular issue in supply chain management. An order-oriented decision is very important for fulfilling the demand of final customers in supply chain management. There is not any research discuss DLSP considering with COP in our reviewing. This paper deals with the dynamic lot-sizing problem considering the order-oriented decision by focusing on the DLSP with COP, namely DLSCOP.

III. PROBLEM FORMULATION

This research focused on the DLSP considering COP, namely DLSCOP. DLSCOP will be defined by a nonlinear programming model in the section. Firstly, there are some assumptions for the DLSCOP should be described as following:

1. The demand is variable and deterministic.
2. Order quantities must be integer multiples of a constant batch size. No other limits are imposed on the size of the order.
3. Cost factors are time independent.
4. Replenishment is instantaneous.
5. Back orders are only allowed to make up for quantity discrepancies that result from batch ordering.
6. An order must be placed in the first period.
7. All items ordered in the same order should be delivered at the same period. Shortages of any item will backorder the entire order.

The following notations are used for developing the model.

c_j: batch size of item j

d_{ijt}: demand of item j of order i at period t.

g: backorder cost per unit per period

h: holding cost per unit per period

M: a large number

p: ordering cost

T: the planning horizon

II_{jt}: independent inventory level of item j at the end of period t,

DI_{jt}: dependent inventory level of item j at the end of period t,

w_{jt}: positive Inventory level of item j at the end of period k. w_{jt} = max(0,II_{jt}+DI_{jt}),

x_{jt}: a decision variable that is a integer multiple of the batch size lending the quantity made or ordered of item j in period t,

y_{jt}: Boolean variable to assign the ordering cost of item j in period t,

IB_{jt}: independent backorder quantity of item j at the end of period t, IB_{jt}= max (0, -II_{jt}),

DB_{jt}: dependent backorder quantity of item j at the end of period t, DB_{jt}=max (0, - DI_{jt}),

z_{jt}: backorder quantity at the end of period t, z_{jt}=max (0, IB_{jt}+DB_{jt}),

y_{jt}: Boolean variable to assign the setup/ordering cost of item j in period t,

u_{ijt}: Boolean variable to assign the dependent inventory level of item j of order i in period t,

v_{ijt}: Boolean variable to assign the dependent backorder of item j of order i in period t.

k_{ijt}: Boolean variable to check d_{ijt} will be fulfilled (not become backorder) or not.

Using the notations above, the mathematical formulation of the problem can be written as follows:

Minimize:

$$\sum_{t=1}^{t}\sum_{j=1}^{n} py_{jt} + hw_{jt} + gz_{jt} \qquad (1)$$

Subject to:

$$II_{j_t} = cx_{jt} + II_{j(t-1)} - d_{jt} \qquad t = 1,\ldots,T \quad j = i,\ldots,n \qquad (2)$$

$$DI_{jt} = \sum_{i=1}^{m} d_{ijt}u_{ijt} \qquad t = 1,\ldots,T \quad j = 1,\ldots,n \qquad (3)$$

$$IB_{jt} < c_j \qquad t = 1,\ldots,T \quad j = 1,\ldots,n \qquad (4)$$

$$IB_{jt} = \max(0, d_{jt} - (W_{j(t-1)} + C_j X_{jt})) \qquad t = 1,\ldots,T \quad j = 1,\ldots,n \qquad (5)$$

$$DB_{jt} = \sum_{i=1}^{m} d_{ijt}v_{ijt} \qquad t = 1,\ldots,T \quad j = 1,\ldots,n \qquad (6)$$

$$My_{jt} \geq x_{jt} \qquad t = 1,\ldots,T \quad j = 1,\ldots,n \qquad (7)$$

$$u_{ijt}, v_{ijt,} y_{jt} \geq (0,1) \qquad t = 1,\ldots,T \quad j = 1,\ldots,n \qquad (8)(9)(10)$$

$$u_{ijt} = 1, \quad if \quad \sum_{j=1}^{n} k_{ijt} < n \qquad i = 1,\ldots,n \quad t = 1,\ldots,T \qquad (11)$$

ISSN (Online) 2456 -1304
International Journal of Science, Engineering and Management (IJSEM)
Volume 11 Issue 8 August 2024

$$v_{ijt} = 1, \ \ if \ \ \sum_{j=1}^{n} k_{ijt} < n \qquad i = 1,\dots,n$$
$$t = 1,\dots,T \qquad (12)$$
$$I_{j0} = 0 \qquad j = 1,\dots,n \qquad (13)$$
$$x_{jt} \geq 0 \qquad t = 1,\dots,T$$
$$j = 1,\dots,n \qquad (14)$$
$$x_{j1} > 0 \qquad j = 1,\dots,n \qquad (15)$$
$$x_{jt} \geq 0 \qquad t = 2,\dots,T$$
$$j = 1,\dots,n \qquad (16)$$
$$w_{jt}, z_{jt} \geq 0 \qquad t = 1,\dots,T$$
$$j = 1,\dots,n \qquad (17)$$

The objective function (1) minimizes total ordering, holding, and backordering cost over time. Constraint (2) models the flow conservation in period *t*. Constraint (3) and (6) model the total of dependent inventory/backorder at period *t*. Constraint (4) limits backorders to compensate for fixing quantities within *c* only. Constraint (5) models the total independent backorder at period t. Constraints (7) and (10) force y_{jt} to zero or one to ensure only charging setup/ordering cost when making an order. Constraints (8), (9), (11), and (12) model the dependent inventory and dependent backorder based on the definition of the customer-ordering problem. Constraint (13) sets the initial inventory level to zero. Constraint (15) ensures making an order in the first period. Finally, constraints (14), (16), and (17) ensure that order, carryover, and backorder quantities are all non-negative.

IV. MODIFIED PARTICLE SWARM OPTIMIZATION

PSO, particle swarm optimization, was first introduced by Kennedy and Eberhart (1995) as an optimization method for nonlinear functions with continuous variables. The initial intention of PSO was to simulate social behavior of flocking birds searching for food by means of exchanging knowledge among flock members. By applying simple formulas, Kennedy and Eberhart developed an optimization algorithm that mimics this knowledge sharing. Each individual in the flock was represented by a point in a two-dimensional space, and future movement of each point in the search space is determined using a combination of previous experience of the individual, and of other individuals in its neighborhood group (Kennedy and Eberhart, 1995). PSO simulates the behavior of birds flocking. If there has a group of birds are randomly searching for food in an area. There is only one piece of food (target) in the area. Not all the birds know where the food is. So what is the best strategy to find the food? The effective one is to follow the bird which is nearest to the food (Hu, 2002).

PSO searches optimal solutions by individual and group experiences; however, the solution of the optimization problem may not come from previous solutions. Certain parameters need to be adjusted and random variables will be put in to distort the optimum solution (Eberhart & Shi, 2001). An advantage of PSO is that these particles remember the best position that they have seen. Members of a swarm communicate better positions to each other and based on this they can adjust their own position and velocity. Schutte and Groenwold (2005) proposed that each agent's search velocity changes as a random function of the distance between a point and a local best, and the distance between the point and the global best.

Each individual in the PSO algorithm is called a "particle". Each particle is subject to a movement in a multidimensional space which it remembers. Particles have memory, and thus would retain part of their previous state. While there are no restrictions for particles to know the positions of other particles in the multidimensional spaces, they can still remember the best positions they have ever had. Each particle's movement is the composite of an initial random velocity, two randomly weighted influences: individuality (the tendency to return to the particle's best previous position), and sociality (the tendency to move towards the neighborhood's best previous position). All of the particles have fitness values which are evaluated by the fitness function to be optimized. Each particle has two main characteristics: velocity and position.

The PSO heuristic described above is applied to continuous optimization. However, because of the wide variety of optimization problems that involve discrete variables, Kennedy and Eberhart (1997) introduced a discrete binary version of the heuristic. In the discrete version of PSO, solutions are represented by a string of binary bits. The velocity of a particular bit is defined as the probability that it will take a value of one. Accordingly, if the velocity is equal to 0.1, then there is a 10% chance of the bit taking a value of one, and 90% chance of it taking a zero value in the next iteration. (Kennedy and Eberhart 1997). This means that each bit in particle is treated separately. Gaafar and Aly (2008) applied PSO to traditional dynamic lot sizing with batch order. In their research, PSO outperformed both the MSM and the GA by producing the lowest cost solution. We will introduce PSO for solving DLSCOP for comparing with GA and traditional methodologies WW and SM.

In the research, a modified PSO (mPSO) has been modeled for solving DLSCOP. A example of binary solution for DLSCOP is presented on figure 3. In figure 2, the example of DLSCOP, the total demand of item 1, 2, 3 are listed for each period. Figure 3 is an example of purchasing decision for the example for item 1, 2, 3. The value of a bit will be 0, 1, and, 2. '1' and '2' represented an order is placed in the corresponding period with a quantity that would satisfy its own demand and the demand of all subsequent periods with a corresponding '0' bit value. The difference between '1' and '2' is related to the batching size. '1' will order more than real demand, namely sufficiency purchasing (SP). '2' will order less than real demand, namely insufficiency purchasing (IP). If the batch size is 6, in period 1, item 1 will purchase for period 1-3. The total demand of item 1 for period 1-3 is 21 (10+8+3=21, figure 2), and the purchasing quantity will be 24 (sufficiency purchasing). The next purchasing decision will make in period 4. It's insufficient purchasing for period 4-6. The total demand is 25 (9+12+4=25, figure 2) and the purchasing quantity will be 24 (6*4=24).

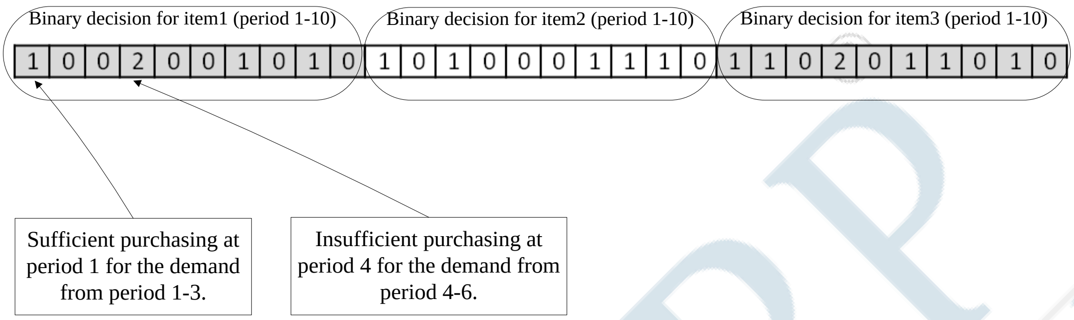

Figure 3. an example of the purchasing decision for DLSCOP

The flowchart for mPSO dealing with DLSCOP is showed on figure 4. Those steps are described following:

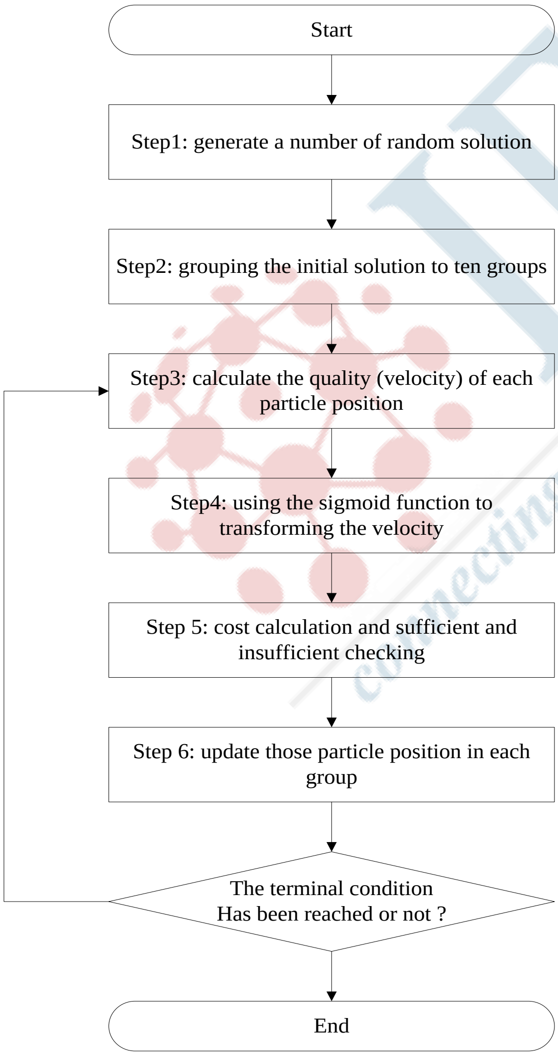

Figure 4. the flowchart of mPSO for DLSCOP

step 1. Generate a number of random solution

mPSO is applied by first generating a number of random solutions (or positions of particles) in the solution space. The solution will be represented by a serial of modified binary string. Each bit in the string represented a single period in a T period planning horizon. Figure 3 is an example of the binary string solution for DLSCOP. It will be different depend on the number of items and periods. There are 100 solutions generated for analyzing in the research.

step 2. Grouping the initial solution to ten groups

In PSO heuristics, the initial solution should be grouped. The group (neighborhood) can be defined in many ways. In the paper, we will use overlapping groups applied in Gaafar and Aly (2008). All the initial solutions will be numbered. Each group will have ten solutions. Solution 1 to 10 assigns to group 1. Group 2 is solution 2-11, and so on.

step 3. Calculate the quality of each particle position (velocity)

The quality of each particle position (velocity) is evaluated based on the object function (total cost). To proceed from iteration *k* to the next iteration *k+1*, velocity of a particle *i* is calculated using the equation (Kennedy and Eberhart, 1997):

$$v_{k+1}^i = v_k^i * (p_k^i - s_k^i) + r_2 * (p_k^g - s_k^i) \quad (1)$$

The new position of the particle is obtained as follows,

$$s_{k+1}^i = s_k^i + v_{k+1}^i \quad (2)$$

v_k^i: Velocity of particle *i* in the current iteration *k*.

p_k^i : The best solution that particle *i* reached throughout iterations *1, 2,…, k.*

p_k^g: The best solution that the group has reached throughout iterations *1, 2,…, k.*

ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)

Volume 11 Issue 8 August 2024

s_k^i:Particle *i* position in the current iteration *k*.

s_{k+1}^i: Particle *i* position in the next iteration *k+1*.

r_1 and r_2: Uniformly distributed random numbers generated between *0* and *1*.

step 4. Using the sigmoid function to transforming the velocity

In the DLSCOP, the variables are discrete binary. The PSO heuristic described above is applied to continuous optimization. For our problem, we will modify the binary version of PSO heuristics introduced by Kennedy and Eberhart (1997). The velocity of a particular bit is defined as the probability that it will take a value of one. Accordingly, if the velocity is equal to 0.1 then there is a 10% chance of the bit taking a value of one, and 90% chance of it taking a zero value in the next iteration *k+1* (Kennedy and Eberhart, 1997). This means that each bit *j* in particle *i* is treated separately. The velocity calculating from the equation at step 3 should be modified for getting the probability in the binary version of PSO. A sigmoid function will use for the transforming described following:

$$sig(v) = \frac{1}{1+e^{-v}} \qquad (3)$$

Where *v* is the velocity calculated from (1).

$$\text{If } r < sig\left(v_{k+1}^{i,j}\right), \text{ then } s_{k+1}^{i,j}=1 \qquad (4)$$

$$\text{Else } s_{k+1}^{i,j}=0$$

Where r is generated from a uniformly distribution between 0 and 1, $v_k^{i,j}$ and $s_k^{i,j}$ represented for the velocity and the position of bit *j* of particle *i* in the iteration *k* separately.

step 5. Cost calculation and SP and IP checking

The cost of those purchasing decision getting from those new positions of those particles generating from step 3-4 will be calculated. And SP or IP check will perform for each bit with the value of 1 (SP). If the cost of IP is less than the cost of SP, the value of the bit will changes from 1 to 2.

step 6. Update those particle position in each group

At step 3-5, new positions of those particles in each group have been updated. The best good one of each group and the best one of all the particles will be updated, too. I will use for calculating the velocity for next iteration.

step 7. Repeat step 3-6 until the terminal condition has been reached

Step 3 to 6 will be repeated for improving the quality of solution. At least 300 iterations will run for each special type of problem.

V. COMPUTATIONAL EXPERIMENTS

The computational experiment considers five factors. Each factor has two values for the evaluation test. The five factors are: item types3, 6, number of orders per period (5, 10), the demand pattern (constant, seasonal), batch size (6, 24), the ratio of the ordering cost to the carrying cost (1, 8), the ratio of the carrying cost to the backorder cost(1,8), and the length of the planning horizon（10, 20）. Demand pattern generated from the model developed from Gaafar & Choueiki (2000). Constant demand pattern model generated from (5):

$$d_{ijt} = \alpha + \varepsilon_{ijt} \quad 1 \le i \le m,\ 1 \le j \le n,\ 1 \le t \le T \ (5)$$

Where, d_{ijt} is the demand in period *t* for order *i* item *j*, T is the number of periods in the plan (factor E), *m* is the number of orders at period *t*, *n* is the number of items, αis a constant generated from an exponential distribution with a mean of 10, and ε_{ijt} is a normally independently distributed error component with a mean of 0 and a constant variance of $\sigma^2 (\sigma = 0.1\alpha)$.

The generated seasonal demand pattern uses the following model:

$$d_{ijt} = a_1 + a_2 + sin\frac{2\pi(t+b)}{T} + \varepsilon_{ijt} \qquad 1 \le i \le m,\ \ 1 \le j \le n,\ \ 1 \le t \le T \qquad (6)$$

Where, α_1 is a constant generated from an exponential distribution with a mean of 10, α_2 is the amplitude of the sinusoidal curve (α_2=0.5α_1), *b* is a constant generated from a discrete uniform distribution ranging between 0 and *T-1* to randomly vary the starting point of the demand pattern, and ε_{ijt} is a normally independently distributed error component with a mean of 0 and a constant variance of σ^2(σ=0.1α_2).

The experiment checks both patterns to ensure that demand in the first period is always greater than zero. This experimentation generates four distinct sets (of two hundred demand patterns each), one for each of the seasonal and constant demand patterns and once for each of the ten and twenty period horizons. Overall, 25600 instants were executed (128 runs times 200 instances per run).

VI. RESULT AND DISCUSSION

Table 1-4 are the cost analysis of the four rules, mPSO, SM, WW, GA, for DLSCOP. We can find the performance of mPSO is better than GA, WW, and SM. Meta-heuristics, GA and mPSO are greater than traditional heuristics WW and SM. mPSO will be better than traditional GA, especially in the complex condition, like as more items type, long period, and seasonal demand. The different of the four methodologies showed on figure 5, too. The relative performance is defined as the cost of mPSO is 1. The relative performance is calculated from the cost of SM, WW, and GA, divided the cost of mPSO. We can find the performance of mPSO is stable than other under different experimental factors. GA can find a superior solution in some condition but the stability is its major weakness. WW is performing better than SM in the stability.

In the performing efficiency of mPSO, WW, SM, and GA, traditional rules (WW, SM) don't need too much time to generate purchasing decision. If we can spend little time to

IFERP

ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)

Volume 11 Issue 8 August 2024

perform meta-heuristic, like as mPSO, the total cost can reduce significantly. In table 5, the performing time of mPSO and GA are comparing. Longest time for GA to have a decision is more than half hour. Performance time of mPSO can be saved more than 50%. Total average of the performance time of mPSO is 133 second is superior than 312 second for GA. Furthermore, we can find the efficiency of mPSO is greater than GA.

Table 1: the cost of mPSO, WW, SM and GA for DLSCOP for run 1 to 32

run	mPSO		SM		WW		GA	
	Average	Std_dev	Average	Std_dev	Average	Std_dev	Average	Std_dev
1	4204.21	243.926	5385.09	218.587	4302.65	251.388	4243.09	249.928
2	4242.13	236.466	5440.33	162.481	4324.64	237.724	4254.26	228.974
3	15008.39	764.452	20827.53	1179.129	15349.59	660.298	15027.65	1008.402
4	15007.87	766.695	20827.09	1179.645	15343.52	658.308	14929.83	937.981
5	8316.80	336.573	10730.08	310.908	8547.76	316.554	8812.06	338.067
6	8449.40	377.844	10936.78	306.491	8693.82	381.734	8762.81	345.827
7	28959.69	1186.702	39666.81	1293.979	29671.65	1451.616	29792.34	1187.411
8	28902.29	1340.240	39624.24	1410.813	29453.30	1593.277	29570.14	1221.089
9	4818.19	247.859	6007.08	193.938	4938.68	260.912	4950.33	243.394
10	4965.26	269.651	6386.45	253.893	5119.23	296.940	4872.84	261.296
11	15673.10	810.271	21408.61	1002.824	16055.09	709.683	15580.52	1008.507
12	15469.35	753.618	21545.07	894.420	15813.10	578.830	15353.95	995.777
13	9685.88	454.624	12218.15	486.572	10010.12	505.504	10694.51	461.826
14	10520.36	610.502	13573.25	643.311	10939.78	666.847	10334.55	429.557
15	30435.18	1396.928	41372.31	1369.864	30983.82	1534.640	31586.60	1334.748
16	29451.52	1516.066	40548.62	1658.355	29980.95	1884.052	30320.15	1586.077
17	4782.14	265.436	5851.69	127.657	4867.96	275.680	4800.63	234.889
18	4731.95	244.167	5806.48	130.222	4822.39	260.929	4806.58	277.230
19	17521.68	1194.885	24357.68	1168.467	17663.24	1225.287	17718.86	1070.192
20	17545.39	1207.804	24326.61	990.735	17652.79	1195.060	17676.14	1211.647
21	9389.70	330.752	11551.99	218.737	9648.57	337.020	10093.84	313.263
22	9415.45	353.552	11620.47	225.259	9631.82	344.242	9978.38	374.420
23	34672.88	1408.450	47341.95	1480.599	35256.46	1464.394	35770.49	1496.546
24	34381.40	1465.590	46930.49	1657.835	35112.50	1500.161	35361.90	1514.410
25	5427.34	223.587	6438.33	147.563	5554.53	235.177	5577.71	303.288
26	5563.35	271.677	6783.07	248.030	5689.18	284.385	5534.48	283.168
27	18358.38	1132.071	24730.72	1071.549	18531.50	1154.644	18495.10	1161.246
28	18019.42	1157.575	24928.44	1054.643	18144.19	1109.625	18128.14	1148.703
29	10694.87	408.402	12942.74	366.492	10921.26	430.345	12092.43	495.523
30	11962.27	536.404	15101.25	643.510	12366.83	580.218	11869.49	408.792
31	35712.83	1460.097	48365.86	1327.875	36518.18	1498.315	37507.50	1524.504
32	35764.09	1540.170	48770.05	1719.380	36587.00	1530.044	36506.22	1669.630

Table 2: the cost of mPSO, WW, SM and GA for DLSCOP for run 33 to 64

run	mPSO		SM		WW		GA	
	Average	Std_dev	Average	Std_dev	Average	Std_dev	Average	Std_dev
33	5338.86	211.748	6106.01	64.488	5398.40	220.509	5423.53	205.864
34	5371.52	215.223	6120.81	63.401	5419.48	212.175	5436.35	229.590
35	20664.06	1209.978	28028.76	1151.579	20899.50	1312.891	20381.04	1236.820
36	20403.93	1123.946	27711.24	1176.635	20703.72	1276.298	20412.56	964.856
37	10620.45	327.693	12171.68	161.529	10750.87	342.670	11346.77	333.957
38	10856.67	305.148	12503.99	156.283	10978.37	280.609	11270.89	318.779
39	39805.83	1419.502	54147.25	1226.796	40557.20	1333.536	41116.07	1394.333
40	40003.82	1536.620	54438.37	1446.888	40898.81	1338.398	41075.29	1674.002
41	5973.88	221.287	6711.45	78.139	6084.91	244.007	6193.80	259.194
42	6225.65	298.528	7232.61	212.645	6323.01	301.958	6141.95	266.227
43	21279.23	1003.634	28514.61	940.640	21491.08	1072.439	21164.64	1116.843
44	21177.76	1325.818	28715.29	1263.797	21464.70	1493.653	20700.09	1180.358
45	12121.70	418.682	13949.10	427.620	12251.78	465.422	13394.80	530.511
46	13545.30	636.906	16591.03	490.610	13812.99	665.521	13109.26	429.128
47	41001.08	1568.610	55487.67	1634.586	41956.47	1447.391	43057.91	1537.334
48	40980.18	1674.724	55964.49	1632.369	41834.85	1644.510	42176.24	1719.645
49	5784.81	156.082	6158.04	37.639	5824.09	152.588	5909.50	193.225
50	5761.98	168.551	6161.55	24.212	5801.78	165.479	5855.12	189.891
51	23919.67	1219.772	32324.16	1290.071	24178.17	1104.478	24358.91	1316.220
52	23860.48	1156.468	32154.20	1218.669	24078.98	1062.657	24219.57	1158.017
53	11504.23	213.440	12295.36	99.232	11608.93	241.948	12405.11	332.677
54	11562.34	226.808	12408.92	101.960	11664.50	233.045	12328.77	285.360
55	46880.92	1824.874	62896.58	1801.293	47763.34	1713.718	49009.11	1548.116
56	47329.80	1624.431	63542.15	1463.083	48270.61	1741.255	48502.61	1631.083
57	6399.09	171.694	6757.17	70.827	6554.44	220.011	6704.26	210.844
58	6668.67	254.259	7194.67	201.130	6775.48	313.594	6693.13	235.049
59	24632.59	1146.013	33198.85	1203.915	24940.35	1097.090	24964.96	1358.360
60	24747.62	1215.121	33443.19	1127.444	24964.19	1156.082	24937.61	1317.752
61	12880.88	334.300	13688.25	303.985	13076.81	369.115	14801.04	465.514
62	15049.37	561.101	16569.42	518.728	15268.21	568.882	14615.80	430.422
63	48329.35	1732.627	64410.61	1797.590	49242.94	1836.464	50810.66	1873.500
64	49072.90	1656.854	65855.42	1511.927	50037.73	1676.394	50375.23	1672.400

Table 3: the cost of mPSO, WW, SM and GA for DLSCOP for run 65 to 96

run	mPSO		SM		WW		GA	
	Average	Std_dev	Average	Std_dev	Average	Std_dev	Average	Std_dev
65	8479.99	490.719	10845.68	389.301	8666.95	483.221	8917.87	401.904
66	8406.00	405.613	10785.74	318.202	8593.16	414.283	8753.12	444.461
67	30147.96	1408.207	41996.63	1616.801	30651.48	1085.451	30863.98	1710.801
68	30150.85	1312.004	41869.58	1768.296	30637.65	1038.876	30525.98	1615.648

IFERP connecting engineers...developing research

ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)

Volume 11 Issue 8 August 2024

69	16594.61	583.168	21382.05	470.851	17115.21	573.816	19314.09	719.480
70	16530.12	592.576	21377.00	495.429	17068.70	573.372	19134.75	683.025
71	58210.60	2351.984	79626.47	3017.750	59153.66	2818.190	76467.52	2808.695
72	58220.10	2351.983	79698.98	2595.591	59161.51	2815.894	71992.01	2228.632
73	9696.82	415.540	12021.45	325.623	9896.22	383.077	10483.73	550.581
74	9898.22	420.662	12261.95	341.095	10111.21	435.278	10282.52	506.079
75	31318.28	1364.303	42676.05	1607.268	32079.32	1111.710	32520.05	1948.538
76	31310.59	1465.641	43026.41	2095.250	32002.49	1229.793	31779.61	1712.183
77	18917.49	624.336	23636.88	510.165	19455.82	628.464	23525.88	1047.718
78	20717.02	794.687	26287.39	990.291	21483.08	849.018	23030.76	847.980
79	60464.55	2293.601	81955.64	2213.240	61975.03	2914.617	81918.30	3199.701
80	60624.61	2553.015	82451.94	2520.821	61947.22	3111.335	76299.63	2752.071
81	9486.74	433.506	11606.57	271.581	9666.92	429.331	9998.38	443.206
82	9538.40	473.930	11666.80	259.750	9709.66	485.316	9978.43	519.277
83	35000.08	2191.047	48377.89	2065.591	35116.26	2127.350	36283.92	1973.224
84	35345.74	2133.958	48688.66	1853.780	35488.98	2139.419	36103.72	2034.572
85	18760.81	738.900	23081.82	563.534	19185.71	740.689	22140.21	865.985
86	18853.83	622.235	23155.54	354.546	19291.77	585.359	21882.68	704.256
87	68677.94	2847.362	94340.63	2737.113	70101.48	2706.656	86103.28	2876.716
88	68671.66	2995.971	93823.38	3076.087	70022.61	3250.350	81827.14	2561.016
89	10777.71	451.798	12863.57	231.641	11014.47	481.728	11796.10	650.083
90	10860.56	480.405	12900.99	294.154	11055.92	473.174	11678.41	517.010
91	36558.13	2047.332	49690.31	2066.744	36797.09	1989.972	37898.20	2130.499
92	36017.46	2077.093	49238.05	2136.697	36270.42	2130.141	37778.30	2141.000
93	21149.86	726.090	25397.22	521.402	21655.86	663.436	26688.11	1199.192
94	22748.05	911.485	27812.98	796.115	23308.38	892.424	26150.99	1046.671
95	71531.55	2661.933	96737.93	2719.638	73115.95	2756.772	91509.11	3897.814
96	71163.41	3201.466	96592.83	3416.654	72955.65	3088.390	86150.29	2813.238

Table 4: the cost of mPSO, WW, SM and GA for DLSCOP for run 97 to 128

run	mPSO		SM		WW		GA	
	Average	Std_dev	Average	Std_dev	Average	Std_dev	Average	Std_dev
97	10783.65	339.395	12223.78	90.772	10881.17	320.862	11311.72	428.952
98	10669.41	372.306	12211.30	102.371	10762.41	362.132	11284.55	437.721
99	40639.00	2232.031	55178.59	2067.912	41058.82	2573.527	41474.21	2166.271
100	41283.20	2146.805	56082.62	1727.063	41826.45	2490.159	41268.90	2033.819
101	21385.53	569.656	24371.96	326.353	21613.23	546.336	24863.67	812.066
102	21339.69	475.732	24417.83	135.166	21580.05	480.275	24495.16	781.375
103	79861.95	2725.942	108282.36	2780.542	81241.21	2479.940	94313.80	2706.804
104	79781.24	2893.840	108280.55	2816.216	81148.57	2574.927	90838.82	2575.422
105	11962.09	409.037	13405.19	119.756	12150.33	454.236	13115.53	548.725
106	12133.66	441.161	13765.26	241.503	12262.61	470.858	13062.17	455.988
107	42449.53	2374.335	56830.20	2112.421	42961.77	2671.267	43620.67	2214.805

108	42577.14	2486.357	57216.73	2336.890	43027.59	2690.258	42932.33	2128.930
109	23581.45	530.813	26634.55	254.952	23860.05	534.574	29675.99	1022.618
110	26256.44	1049.996	30967.13	724.652	26574.94	1071.833	29060.22	912.005
111	81772.37	2835.619	110142.64	2939.330	83513.21	2659.112	99877.99	3474.340
112	83319.88	3084.459	111737.69	5394.320	84881.69	2802.518	94631.78	2992.805
113	11511.98	312.010	12308.42	86.861	11562.12	304.754	12423.51	542.170
114	11511.98	312.010	12308.42	86.861	11562.12	304.754	12327.58	479.146
115	47746.64	2272.960	64462.94	2495.828	48174.92	2186.466	49666.91	2384.461
116	47746.64	2272.960	64462.94	2495.828	48174.92	2186.466	48939.06	2343.422
117	22982.72	417.431	24602.52	48.060	23173.28	423.047	29217.27	1393.134
118	23059.07	415.644	24608.48	42.101	23219.39	452.289	28100.29	1033.896
119	94756.64	3517.391	126964.77	3654.952	96510.30	3454.178	108270.73	3201.210
120	94540.40	2947.673	126543.72	4274.487	96212.11	2966.632	105036.25	3109.847
121	12747.13	297.808	13503.61	96.892	13010.25	363.804	14425.46	752.796
122	12822.53	338.602	13558.66	111.007	13133.59	436.065	14135.12	541.337
123	49309.46	2078.178	65701.36	2429.791	49810.34	2026.629	51295.95	2458.176
124	49337.67	2213.155	66050.06	2144.137	49830.56	2015.188	50481.63	2616.960
125	25279.22	460.097	26842.00	239.240	25596.62	565.442	34056.11	1487.370
126	27582.90	818.832	29759.80	773.725	27860.61	855.753	32853.17	1132.294
127	96382.54	3382.199	128446.03	3086.746	98145.58	3332.065	112712.37	3637.629
128	97948.74	3374.596	130515.11	3134.650	100048.98	3023.383	109285.52	3231.317

Table 5: the performing time of mPSO and GA

runs	GA_time	mPSO_time	runs	GA_time	mPSO_time	runs	GA_time	mPSO_time	runs	GA_time	mPSO_time
1	00:59	00:46	33	01:04	00:16	65	02:27	00:40	97	04:54	00:59
2	01:05	00:53	34	01:13	00:19	66	02:33	00:46	98	05:14	01:12
3	00:43	00:35	35	00:55	00:11	67	02:10	00:35	99	04:01	00:44
4	00:48	00:08	36	01:08	00:13	68	02:19	00:36	100	03:24	00:50
5	03:29	00:40	37	03:29	01:15	69	05:45	03:35	101	08:49	05:53
6	05:15	00:58	38	05:56	01:58	70	06:43	04:17	102	11:27	07:19
7	02:09	00:31	39	02:21	00:53	71	04:53	02:45	103	06:31	04:05
8	02:53	00:43	40	03:55	01:21	72	04:30	03:10	104	07:33	04:44
9	01:05	00:10	41	01:12	00:18	73	02:02	00:44	105	02:53	01:08
10	01:30	00:18	42	03:15	00:35	74	02:48	01:09	106	06:57	01:51
11	00:43	00:08	43	00:58	00:13	75	01:49	00:36	107	02:39	00:52
12	01:28	00:13	44	02:27	00:23	76	02:39	00:49	108	04:49	01:14
13	02:16	00:52	45	03:50	01:40	77	05:10	03:53	109	10:19	06:34
14	05:29	02:21	46	24:37	05:04	78	25:51	08:19	110	26:26	18:12
15	01:56	00:38	47	02:44	01:15	79	04:20	02:39	111	06:48	05:23
16	05:53	01:40	48	14:19	03:21	80	10:13	05:59	112	28:17	13:02
17	01:03	00:09	49	01:09	00:16	81	02:05	00:33	113	03:42	01:04

18	01:06	00:10	50	01:16	00:19	82	02:38	00:38	114	03:59	01:42
19	00:55	00:07	51	00:56	00:13	83	02:12	00:28	115	02:49	00:59
20	01:00	00:08	52	01:01	00:14	84	02:19	00:30	116	03:01	00:52
21	02:09	00:38	53	03:10	01:18	85	06:13	02:43	117	10:57	06:09
22	03:15	00:48	54	05:04	01:45	86	06:48	03:20	118	13:44	07:30
23	02:06	00:30	55	02:50	00:54	87	04:57	02:11	119	05:49	04:20
24	02:30	00:37	56	03:47	01:11	88	05:11	02:25	120	06:36	04:59
25	01:00	00:10	57	01:10	00:18	89	02:36	00:37	121	03:22	01:08
26	01:42	00:15	58	02:43	00:30	90	02:55	00:46	122	04:19	01:30
27	00:55	00:08	59	00:57	00:13	91	02:16	00:31	123	02:44	00:51
28	01:11	00:11	60	01:56	00:21	92	03:42	00:34	124	03:20	01:01
29	02:16	00:46	61	04:29	01:35	93	05:42	03:04	125	12:07	08:33
30	08:37	01:58	62	25:45	04:24	94	16:36	06:28	126	34:42	19:17
31	01:57	00:35	63	02:38	01:10	95	05:12	02:33	127	10:25	06:35
32	05:58	01:20	64	09:15	02:58	96	10:25	04:36	128	17:52	13:04
									average	05:12	02:13

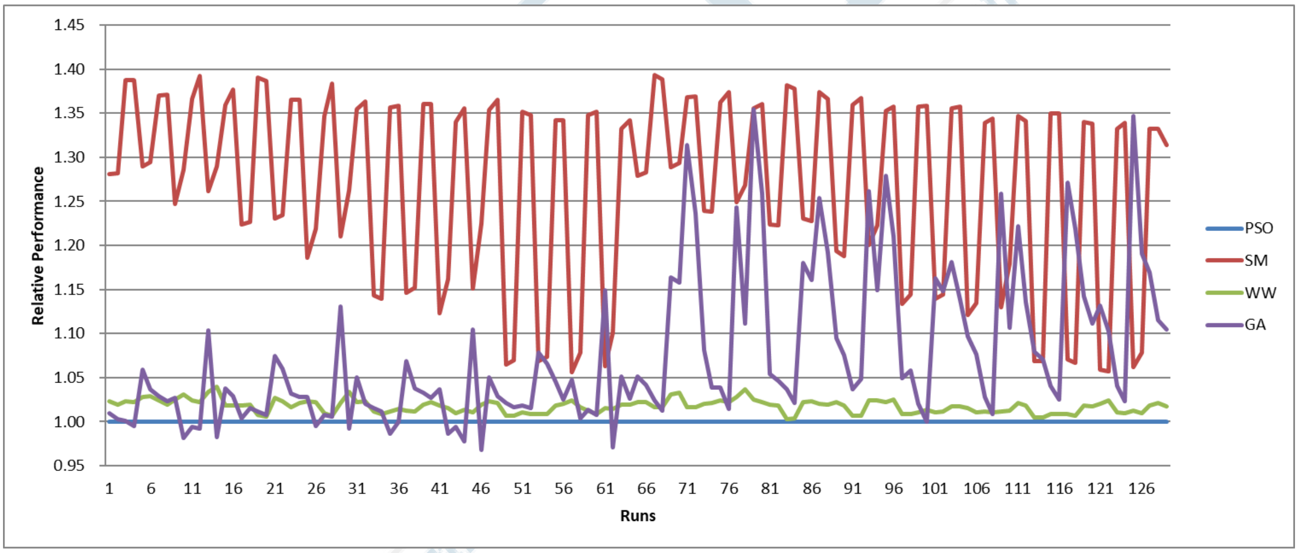

Figure 5. the relative performance of mPSO, SM, WW, and GA for DLSCOP

REFERENCES

[1] Adetunji, O.A.B., Yadavalli, V.S.S., 2012, "An integrated utilization, scheduling and lot-sizing algorithm for pull production," International Journal of Industrial Engineering, 19(32), 171-180.

[2] Ahmadi R., Bagchi U. & Roemer T. A., 2005, Coordinated scheduling of customer orders for quick response, Naval Research Logistics, 52, 493-512.

[3] Bahl, H.C., Ritzman, L.P., HGupta, J.N.D., 1987, Determining lot sizes and resource requirements: A review, Operations Research, 35(3), 329-345.

[4] Baker K. R., 1988, Scheduling the production of components at a common facility, IIE Transactions, 20, 32-35.

[5] Basnet, C., & Leung, J.M.Y., 2002, Inventory lot sizing with supplier selection, In proceedings of the 37th ORSNZ Conference, New Zealand, University of Auckland.

[6] Blocher J. D., Chhajed D. & Leung M., 1998, Customer order scheduling in a general job shop environment, Decision Sciences, 29, 951-981.

[7] Coffman, E. G., Nozari A. & Yannakakis M., 1989, Optimal scheduling of products with two subassemblies on a single machine," Operation Researches, 37, 426-436.

[8] Daganzo C. F., 1989, The Crane scheduling problem, Transportation Research Part B, 23, 159-175.

[9] De bodt, M.A., Gelders, L.F., Van Wassenhove, L. N., 1984, Lot sizing under dynamic demand conditions: A reviewer,

Engineering Costs and Production Economics, 8, 165-187.
[10] Eberhart, R. C., & Shi, Y., 2001, Particle swarm optimization: developments, applications and resources, Proceedings of Congress on Evolutionary Computation, 1, 27-30.
[11] Florian, M., Lenstra, J. K., Rinnooy, Kan, A. H. G., 1980, Deterministic production planning: Algorithms and complexity, Management Science, 26(7), 669-679.
[12] Gaafar, L.K., & Aly, A.S., 2008, Applying particle swarm optimization to dynamic lot sizing with batch ordering, International Journal of Production Research, iFirst, 1-17.
[13] Gaafar, L. K., Choueiki, M. H., 2000, A neural network model for solving the lot-sizing problem, Omega, 28, 175-184.
[14] Gerodimos A. E., Glass C. A. & Potts C. N., 2000, Scheduling the production of two-component jobs on a single machine, European Journal of Operational Research, 120, 250-259.
[15] Gupta, J. N. D., Ho, J. C. & van der Veen A. A., 1997, Single machine hierarchical scheduling with customer orders and multiple job classes, Annuals of Operations Research, 70, 127-143.
[16] Hsu, S. Y., Liu, C. H., 2009, Improving the Delivery Efficiency of the Customer Order Scheduling Problem in a Job Shop, Computers & Industrial Engineering, 57, 856-866 DOI information:10.1016 /j.cie. 2009 .02. 015
[17] Hu, X. H., 2002, PSO Tutorial, Retrieved April 2, 2010 from the World Wide Web: http://www.swarmintelligence.org/ tutorials.php
[18] Hung, Y., & Chien, K., 2000, A multi-class multi-level capacitated lot sizing model, Journal of the Operational Research Society, 51, 1309-1318.
[19] Holland, J., 1975, Adaptation in natural and artificial systems. Ann Arbor, Michigan: The University in Michigan Press.
[20] Jans, R., & Degraeve, Z., 2007, Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches, European Journal of Operation Research, 177, 1855-1875.
[21] Julien, F. M. & Magazine, M. J., 1990, Scheduling customer orders: An alternative production scheduling approach, Journal of Manufacturing and Operations Management, 3, 177-199.
[22] Kennedy, J., & Eberhart, R.C., 1995, Particle swarm optimization, Proceedings of the 1995 IEEE International Conference on Neural Networks, Perth, Australia, 1942-1948.
[23] Kennedy, J., & Eberhart, R.C., 1997, A discrete binary version of the particle swarm optimization algorithm, Proceeding of the 1997 Conference on System, Man, and Cybernetics (SMC'97), 4104-4109.
[24] Khouja, M., Michalewicz, Z., & Wilmot, M., 1998, The use of genetic algorithms to solve the economic lot sizing scheduling problem, European Journal of Operational Research, 110, 509-524.
[25] Kim, B. S., Lee, W. S., 2013, "A multi-product dynamic inbound ordering and shipment scheduling problem at a third-party warehouse," International Journal of Industrial Engineering, 20(1-2), 36-46.
[26] Moon, I., Silver, E., & Choi, S., Hybrid genetic algorithms for the economic lot scheduling problem, International Journal of Production Research, 40, 809-824.
[27] Ozdamar, L., & Birbil, S., 1998, Hybrid heuristics for the capacitated lot sizing and loading problem with setup times and overtime decision, European Journal of Operational Research, 110, 525-547.
[28] Peterkofsky R. I. & Daganzo C. F., 1990, A branch and bound solution method for the crane scheduling problem, Transportation Research Part B, 24B, 159-172.
[29] Prasad, P., & Krishnaiah Chetty, O., 2001, Multilevel lot sizing with a genetic algorithm under fixed and rolling horizons, International Journal of Advanced Manufacturing Technology, 18, 520-527.
[30] Robinson, P., Narayanan, A., Sahin, F., 2009, Coordinated deterministic dynamic demand lot-sizing problem: A review of models and algorithms, Omega, 37, 3-15.
[31] Sarker, R., & Newton, C., 2002, A genetic algorithm for solving economic lot size scheduling problem, Computers and Industrial Engineering,
[32] Schutte, J., & Groenwold, A., (2005, A study of global optimization using particle swarms, Journal of Global Optimization, 31(1), 93-108.
[33] Shittu, E., 2003, Applying genetic algorithms to the deterministic time-varying fixed quantity lot sizing problem, Masters Thesis, Cairo, Egypt: The American University in Cairo.
[34] Silver,E.A. & Peterson, R., 1985, Decision systems for inventory management and production planning, 2nd ed., John Wiley, N. Y.
[35] Staggemeier, A., Clark, A., Aickelin, U., & Smith, J., 2002, A hybrid genetic algorithms to solve a lot sizing and scheduling problem, In Proceedings of the 16th triennial conference of the international federation of operational research societies, Edinburgh, U.K.
[36] Wagner, H.M. & Whitin, T.M., 1958, Dynamic version of the economic lot size model, Management Science, 5(1), 89-96.
[37] Wang G. & Cheng T.C.E., 2007, Customer order scheduling to minimizing total weighted completion time, Omega, 35, 623-26.
[38] Yang J. & Posner A.E., 2005, Scheduling Parallel Machines for the Customer Order Problem, Journal of Scheduling, 8, 49-74.
[39] Yang, J., 2005, "The Complexity of Customer Order Scheduling Problems on Parallel Machines, Computers & Operations Research, 32, 1921-1939.