Particle Swarm Optimization for Yard Truck Scheduling in Container Terminal with a Cooperative Strategy

Ben Niu, Fangfang Zhang, Li Li and Lang Wu

Abstract Trucks play a significant role in transporting containers between the seaside and storage yard at a container terminal. This paper exhibits a cooperative strategy for scheduling trucks, which allows trucks working or acting together toward a common purpose that can reduce truck-unload rate and cut back the make span. The objective is to minimize the total time cost of the sum of the delay of requests and the travel time of yard trucks. Particle swarm optimization (PSO) algorithm and three of its variants are applied to deal with the scheduling problem. The effectiveness of PSOs are analyzed by four typical different level-scale test problems. The results demonstrate that social learning PSO (SLPSO) can obtain better results than other algorithms for different scale cases.

Keywords Yard truck scheduling · Cooperative scheduling · Particle swarm optimization

1 Introduction

Due to the world trade expansion, container traffic has been growing steadily and this trend is expected to continue. This calls for efficient container terminal operations. Therefore, the optimal management for container terminals is desperately needed.

There are three fundamental equipment in typical container terminals: quay cranes (QCs), yard trucks (YTs) and yard cranes (YCs) [1]. When a vessel arrives at a port, containers are discharged by QCs. And then, YTs are utilized to transport the containers to the storage yard. The storage yard refers to the area where containers handling, transport, storage and transfer are occurred. YTs play a significant role in the process of transportation between the seaside and storage yard.

B. Niu (✉) · F. Zhang · L. Li · L. Wu
College of Management, Shenzhen University, Shenzhen, China
e-mail: Drniuben@163.com

G. Leu et al. (eds.), *Intelligent and Evolutionary Systems*,
Proceedings in Adaptation, Learning and Optimization 8,
DOI 10.1007/978-3-319-49049-6_24

The yard truck scheduling problem has characteristics of backhaul due to the repeated pickup and delivery processes involved. In traditional trucks scheduling in container terminals, the itinerary of a single-truck consists of three major steps. Firstly, a truck goes to a quay crane in discharging (called as CD) to pick up a container. Secondly, it delivers the container to an assigned storage area (called as AD). Finally, it turns back to the original CD. This is a typically static scheduling between the seaside and storage yard [1]. For more dynamic scenario, for example, a truck may go to another CD to discharge containers or a storage area for export containers (called as AL) to load a container after the completion of previous delivery operation.

There are many studies on the yard truck scheduling problem. In [2], a mixed-integer programming model was raised to decide the yard truck fleet size and allocate delivery jobs to YTs. In [3], the time-space network technique was utilized to describe the potential movements of yard trucks, thus to decrease the pollution of yard trucks operations. Taking the yard truck scheduling and storage allocation as a whole, Wang et al. [4] studied the influence of yard truck configurations on the truck employment strategy. In [5], a dynamic truck scheduling model with strong applicability was designed to reduce truck-load rate, and shorten the time of handling task. Wang et al. [6] gave weight to both internal truck scheduling and storage allocation and put forward a model that determines the strategy of owning and renting trucks in container terminals. Emphasizing on sequence-dependent processing time and different preparation time, the problem of scheduling a fleet of trucks to perform a set of transportation jobs was investigated in [7].

Unfortunately, most of the research above is not directly applicable to container terminal operations while disregarding their dynamic nature. Yet, as we mentioned before, on most dynamic scenario, container terminal operations need greater flexibility. And the development of models should take into account the characteristics and constraints associated with container terminals.

Taking dynamic characteristics into consideration, this paper addresses the truck scheduling problem in the container terminal using cooperative scheduling strategy, where trucks are normally considered to load a container to the assigned quay crane in AL for export containers in loading operation (called as CL) after the delivery to AD. Figure 1 is a truck-map which shows the cooperative scheduling of trucks in the terminal.

As can be seen from Fig. 1, truck 1 turns to the AL (33) for export container to load a container after delivering its container in AD (23) while truck 2 goes to the AL (25) for export container to load a container after handing over its container in AD (15). Obviously, this cooperative scheduling strategy will greatly reduce the unload time compared with traditional static scheduling strategy.

Scheduling problem is NP-hard [8]. Heuristic algorithms have been widely applied to deal with this kind of problem and achieve more effective solutions. Lee et al. [9] used the preparation time for jobs as the representation of the chromosome, instead of using job sequence which is generally employed in the typical genetic algorithm. Chung et al. [10] presented a new hybrid genetic algorithm with exhaustive searching in order to achieve fine local searching to determine the

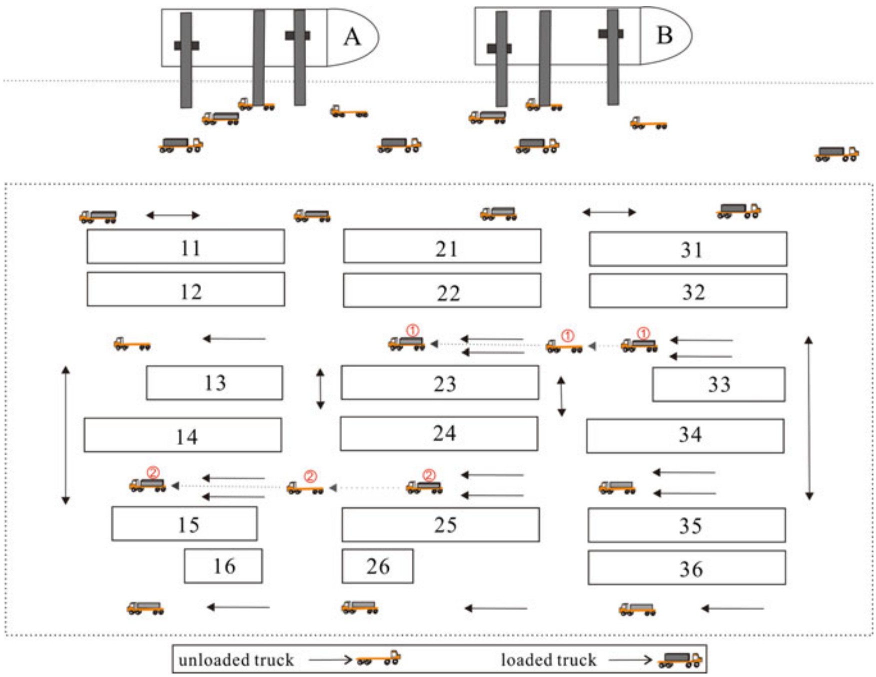


Fig. 1 An overview of cooperative scheduling for trucks

production schedule in the factories. Niu et al. [11] designed a mapping schema using bacterial foraging optimization to deal with integrated yard truck scheduling and storage allocation.

Motivated by the foraging behavior of birds, PSO algorithm was developed by Kennedy et al. in 1995 [12]. Then many researchers have studied the mechanism of PSO algorithm and proposed variety of its variants, such as [13–15]. Particle swarm optimization algorithm, as a significant branch of swarm intelligence, is also widely used in many application fields. In order to dig deeper in PSO algorithm and evaluate its performance in scheduling problems, this paper adopts PSOs to verify the model we proposed.

This study optimizes the yard truck scheduling in container terminal with the cooperative strategy. In the next section we describe the problem in detail. The scheduling problem is formulated in Sect. 3. The PSOs for solving the scheduling problem are discussed in Sect. 4. The results of computational experiments are presented in Sect. 5. The final section concludes the paper.

2 Problem Description

Aiming at improving YTs productivity, more dynamic itinerary courses should be considered. Three dynamic itinerary routes were mentioned in [1], as follows.

- pick up containers at a CD, deliver them to ADs, and go to another CD
- after moving containers from a CD to ADs, go to ALs
- after loaded at a CD, go to ADs and proceed to a CL

In this paper, we focus on the third itinerary course and elaborate the scenario. We assume at least four jobs should be done. The following example includes three quay cranes and four container storage points (two storage points for discharging containers and two storage points for loading containers). Import containers discharged by crane A, B are assigned to storage area AD_1 and AD_2, respectively. And export containers stored at AL_1, AL_2 are assigned to crane B and C for their loading, respectively. The dotted line represents the unloaded itinerary while the solid line stands for loaded itinerary.

Figure 2 illustrates the process of traditional static handling operation in container terminal. In the process of discharging, trucks turn back to their original location (unloaded) after finishing the unloading jobs. Similarly, in the process of loading, trucks have to go to the Ads (unloaded), to pick up export containers and transfer them to assigned QCs. This scheduling strategy will leave the trucks unloaded frequently.

The cooperative scheduling addressed in this paper is defined with a given set of jobs, as shown in Fig. 3. When a vessel arrives at the terminal, containers are discharged on trucks by the QCs. For example, a truck transfers the container to the assigned storage area AD_1. Then the truck moves to AL_1 to pick a container for export. So, the container picked up from AL_1 will be transported to the assigned QC incidentally. However, in a busy terminal, many containers are waiting to be transported nearby the quay crane B, so the truck can undertake one job (for example, transfers a container to AD_2). The same rule is true for the following scheduling. The truck moves to AL_2 to pick up a container for export. And finally, the truck stops nearby quay crane C and waits for a new job. Obviously, this cooperative scheduling strategy can decrease the rate of unloaded itinerary to a large extent.

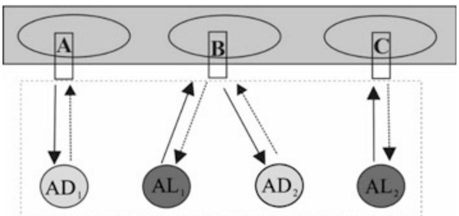


Fig. 2 Before cooperative scheduling

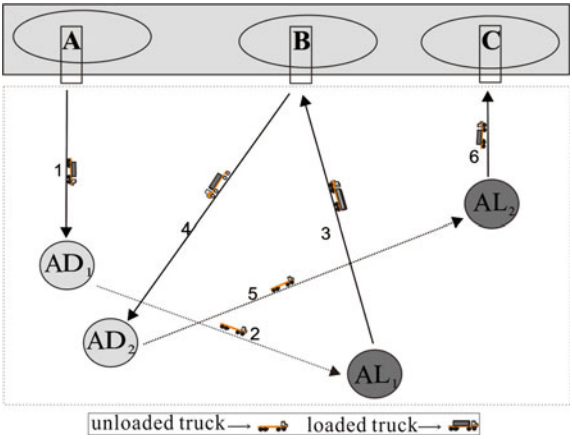


Fig. 3 After cooperative scheduling

As described above, the cooperative scheduling strategy we proposed is more advantageous than the traditional static one.

3 Model Development

In this section, we build a model to minimize the total time cost, that is, the sum of the delay of requests and the travel time of yard trucks. And the flowchart of the cooperative scheduling strategy is shown in Fig. 4.

3.1 Modeling Assumptions

According to the previous research on yard truck scheduling [8, 16], the following assumptions are made in this study.

- The number of trucks is limited.
- The number of storage locations is no less than the number of discharging containers.
- The quay crane and yard crane are always available. That is to say, once the yard truck arrives at the quay crane or yard crane, it can be served immediately.
- Congestion among yard trucks is not considered.
- The pick-up and drop-off locations of each job are known and uniquely identified by their (x, y) coordinates.
- The truck travel speed is the same for both loaded and empty trips.

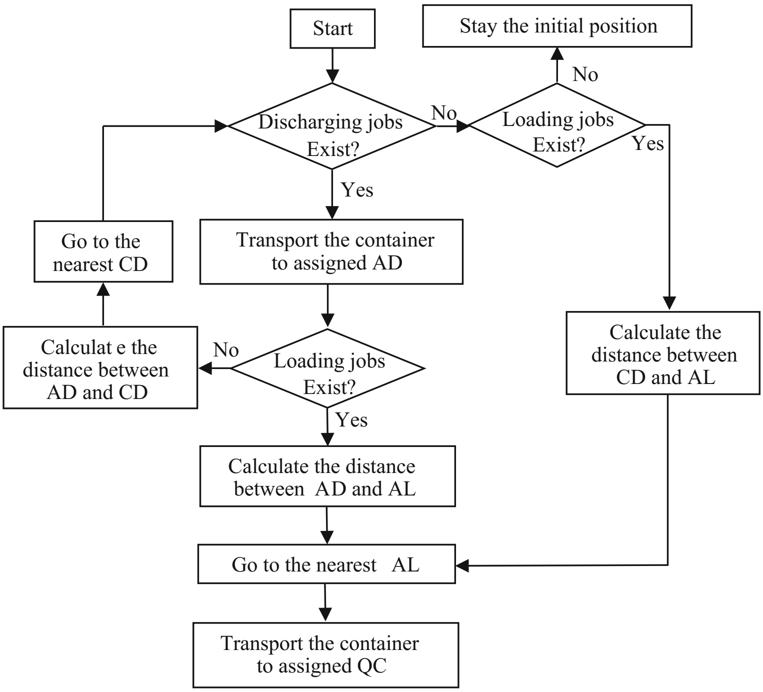


Fig. 4 The flowchart for the cooperative scheduling strategy

3.2 Notations

The following notations are used in this study.

i, j	Index of jobs, $i \neq j$
p, q	Index of location
N	The number of containers to be transported
M	The number of trucks to be used
J^-	Set of discharging jobs
J^+	Set of loading jobs
$J^{'}$	Union set of all jobs and initial status, $J^{'} = J \cup \{l_r\}$
$J^{''}$	Union set of all jobs and final status, $J^{''} = J \cup \{k_r\}$
w_i	Starting time of job i
$[a_i, b_i)$	A soft time window for each job. Ii is a period of time involving the earliest possible a_i and the due time b_i
c_i	Completion time of request i.
d_i	Delay of request i. $d_i = max\{0, c_i - b_i\}$.
$\tau_{p,q}$	The travel time between location p and location q
t_i	The processing time of job i

	$\tau_{oi,ei}$, if job i is a loading job.
	$\tau_{oi,\zeta k}$, if job i is a discharging job and allocated to storage location k.
L_m	The initial location for a truck.
r_m	The initial departure time for a truck
P_i	The pick-up location of job i
Q_i	The drop-off location of job i.
$S_{i,j}$	Setup time of trucks from the destination of job i to the origin of job j.
	τ_{ei}, o_j, if job i and job j are discharging jobs.
	$\tau_{ei,\zeta j}$, if job i is a discharging job and job j is a loading job.
x_{ik}	1, if container i is allocated to storage location k.
	0, otherwise.
X_{ijm}	1, if truck m ($\forall m \in M$) processes job j after job i.
	0, otherwise.
Y_{im}	1, if truck m ($\forall m \in M$) processes job i ($\forall i \in N$).
	0, otherwise.

3.3 Model Formulation

In the process of cooperative scheduling, we aim at decreasing the unloaded itinerary rate to minimize the total make span of the transportation jobs. The processing time of job i has two components: the travel time for the empty trip to P_i (if there is any) and the complete time of job i. The problem formulation is modified based on the model provided by Ng et al. [17] and Lee et al. [8]. However, Ng et al. [17] proposed to schedule a fleet of trucks to perform a set of discharging jobs, ignoring the loading jobs. And Lee et al. [8] only considered the typical static operation pattern in the process of scheduling. In this paper, we take the loading jobs and dynamic itinerary routes into consideration. The revised model is given as follows:

$$Minimize\colon Z = \alpha_1 \sum_{i \in N} d_i + \alpha_2 \left(\sum_{i \in N} t_i + \sum_{i,j \in J} s_{ij} X_{ijm} \right) \quad (1)$$

$$\sum_{m=1}^{M} Y_{im} = 1 \quad \forall i \in N \quad (2)$$

$$\sum_{j=1 \quad and \quad i \neq j}^{N+1} X_{ijm} \leqslant Y_{im} \quad \forall i \in N \quad (3)$$

$$\sum_{j=1 \quad and \quad i \neq j}^{N} X_{ijm} \leqslant Y_{im} \quad \forall i \in N \quad (4)$$

$$1 \geqslant X_{ijm} + X_{ijm} \geqslant Y_{im} + Y_{jm} - 1 \quad \forall i, j \in N \quad and \quad i \neq j; \quad \forall m \in M \tag{5}$$

$$w_i + S_{ij} + t_i \leqslant K(1 - X_{ijm}) + w_j \quad \forall i \in J^{'} \quad and \quad \forall i \in J^{''} \tag{6}$$

$$a_i + d_i \leqslant c_i \quad \forall i \in N \tag{7}$$

$$r_m + t_{L_m, P_i} + d_i \leqslant c_i \quad \forall i \in N; \quad \forall m \in M \tag{8}$$

$$s_{ij} = \tau_{e_i, o_j} \quad \forall i, j \in J^- \tag{9}$$

$$s_{ij} = \sum_{k \in K} \tau_{e_i, \zeta_j} x_{jk} \quad \forall i \in J^- \quad and \quad \forall j \in J^+ \tag{10}$$

$$t_i = \tau_{o_i, e_i} \quad \forall \in J^+ \tag{11}$$

$$t_i = \sum_{k \in K} \tau_{o_i, \zeta_k} x_{ik} \quad \forall i \in J^- \tag{12}$$

$$x_{ik}, X_{ijm}, Y_{im} \in \{0, 1\}, \forall i \in J^{'}, \forall i \in J^{''} \quad and \quad \forall k \in K \tag{13}$$

$$X_{ijm}, Y_{im} \in \{0, 1\}, \quad \forall i \in N - 1; \forall i \in N \quad \forall m \in M \tag{14}$$

$$w_i \in R \quad \forall i \in J^{'} \cup J^{''} \tag{15}$$

$$t_i \in R \quad \forall i \in J \tag{16}$$

$$S_{ij} \in R \quad \forall i \in J \quad and \quad \forall j \in J \tag{17}$$

$$d_i \geqslant 0 \quad \forall i \in J^{'} \cup J^{'} \tag{18}$$

The movement of a container from its origin to destination is defined as a job, denoted by *i* and *j*. Two types of jobs are considered in this paper, loading jobs and discharging jobs. Let J^+ and J^- represent the set of loading jobs and the set of discharging jobs, respectively. A soft time window $[a_i, b_i)$ for each job is given as a constant.

Constraint (2) states that each job is processed by only the same truck. Constraints (3)–(5) give the relationship between *X* and *Y* for jobs handled by the same truck. Constraint (6) gives the connection of the starting time of a job and that of its successor. Constraint (7) defines the relationship between the completion time, preparation time and duration of a job. Constraint (8) gives the relationship between the duration and completion time, the truck preparation time and the travel time of a truck from its initial location to the pick-up location. Constraints (9) and (10) define the setup time of trucks from the destination of job *i* to the origin of job *j*. Constraints (11) and (12) define the processing time of job *i*. Constraints (13)–(18) are simple constraints which define the range of values of some variables.

4 Solution Approach

This paper aims at employing PSOs in solving the scheduling problem. Tasgetiren et al. [18] presented the completion time for n-job m-machine problem by equations. Similarly, in the yard truck scheduling problem, given the processing times t_{im} for job i on truck m $(k = 1, 2, \ldots, m)$, and a job permutation $\pi_i = [\pi_{i1}, \pi_{i2}, \ldots, \pi_{in}]$, n jobs $(J = 1, 2, \ldots, n)$ will be sequenced through m trucks. Let $T(\pi_J, m)$ denotes the completion time of job π_J on truck m. The calculation of completion time for n-job m-truck problem is given as follows:

$$T(\pi_1, 1) = t_{\pi 1, 1} \tag{19}$$

$$T(\pi_J, 1) = T(\pi_{J-1}, 1) + t_{\pi J, 1} \quad J = 2, 3, \ldots n \tag{20}$$

$$T(\pi_J, k) = T(\pi_{J-1}, k) + t_{\pi J, k} \quad J = 2, 3, \ldots n, \quad k = 2, 3, \ldots, m \tag{21}$$

Then the yard truck scheduling is to find a permutation $\pi*$ in the set of all permutations $\prod$. As demonstrated by the following equation.

$$T(\pi*) \leqslant T(\pi_n, m) \quad \forall \pi \in \prod \tag{22}$$

4.1 Particle Swarm Optimization Algorithms (PSOs)

After Kennedy proposed particle swarm optimization in 1995 [12], inertia weight was introduced into PSO algorithm (called SPSO) to provide a balance between global and local exploration abilities by Shi et al. in 1998 [19]. The key optimization mechanism of SPSO algorithm is described as follows.

$$V_{id} = w * V_{id} + c_1 * rand() * (P_{id} - X_{id}) + c_2 * rand()(P_{gd} - X_{id}) \tag{23}$$

$$X_{id} = X_{id} + V_{id} \tag{24}$$

The vector $P_{id} - X_{id}$ represents the distance from individual's current positions (X_{id}) to the individual's previous best position (P_{id}). The $G_{gd} - X_{id}$ indicates the distance between current positions (X_{id}) and the best position (P_{gd}) that has been found by any member of the neighborhood. c_1 and c_2 are two positive constants, $rand()$ is a uniformly distributed random function in the range [0, 1], and w is the inertia weight.

Three improved PSOs, including CLPSO, LPSO and SLPSO can be referred to literature [13–15], respectively. The main updating equations of the three PSOs are described in Table 1.

Table 1 The chosen PSOs for comparison

Algorithm	Updating equations
CLPSO	$V_{id} = w * V_{id} + c * rand() * (P_{id} - X_{id})$
LPSO	$V_{id} = w * V_{id} + c_1 * rand() * (P_{id} - X_{id}) + c_2 * rand()(P_{gd} - X_{id})$ *with topology structure of Square*
SLPSO	$\Delta X_{ij}(t+1) = r_1 * \Delta X_{ij}(t) + r_2(t) * I_{ij}(t) + r_3 * \epsilon * C_{ij}(t)$ $with\, I_{ij}(t) = X_{kj}(t) - X_{ij}(t), C_{ij}(t) = X_j(t) - \overline{X}_{ij}(t)$

PSOs are used to solve difficult continuous optimization problems. However, the scheduling problem in this paper is a discrete one. So, we have to enable the PSOs to be applicable to the continuous problem.

4.2 Solution Representation

In our former research, solution representation was discussed in [20]. The position vector of each particle $\pi_{i(n^+ + n^- + l + m)}$ with $(n^+ + n^- + l + m)$ dimensions is divided into three parts. The $(n^+ + n^-)$ dimensions $\pi_i^J = [\pi_{i1}, \pi_{i2}, \ldots, \pi_{in}(n^+ + n^-)]$ denote scheduling permutation of jobs. The l dimension $\pi_i^l = [\pi_{i(n^+ + n^- + 1)}, \pi_{i(n^+ + n^- + 2)}, \ldots, \pi_{i(n^+ + n^- + l)}]$ denotes potential locations available to the discharging containers. We distinguish them from job permutation part with negative numbers. The m dimension $\pi_i^m = [\pi_{i(n^+ + n^- + l + 1)}, \pi_{i(n^+ + n^- + l + 2)}, \ldots, \pi_{i(n^+ + n^- + l + m)}]$ denotes workload assignment, namely, the number of jobs assigned to each truck.

In order to design a corresponding relationship between the scheduling problem and the particles, a suitable mapping to convert continuous position of particles $X_i^J = [X_{i1}, X_{i2}, \ldots, X_{i(n^+ + n^-)}]$ into job sequence $\pi_i^J = [\pi_{i1}, \pi_{i2}, \ldots, \pi_{in(n^+ + n^-)}]$ in PSOs is needed. The smallest position value (SPV) rule [18] is employed in this study.

Table 2 exhibits the solution representation of particle X_i^J for PSOs with its corresponding sequence. According to the SPV rule, the smallest position value is -1.57, so the dimension $j = 5$ is assigned to be the first job in the processing sequence; the second smallest position value is 0.03, so the dimension $j = 2$ is assigned to be the second job in the processing sequence, and so on.

Table 2 Solution representation of particle X_i

j	1	2	3	4	5	6
X_i^J	2.34	0.03	3.13	0.78	–1.57	1.87
π_i^J	5	2	6	3	1	4

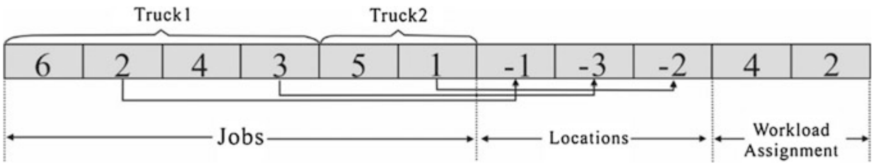


Fig. 5 An example for encoding scheme

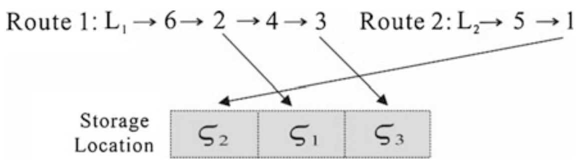


Fig. 6 Decoding of encoding scheme illustrated in Fig. 5

The l dimension $\pi_i^l = [\pi_{i(n^+ + n^- + 1)}, \pi_{i(n^+ + n^- + 2)}, \ldots, \pi_{i(n^+ + n^- + l}]$ is the permutation of the integers from -1 to L. The m dimension $\pi_i^m = [\pi_{i(n^+ + n^- + l + 1)}, \pi_{i(n^+ + n^- + l + 2)}, \ldots, \pi_{i(n^+ + n^- + l + m)}]$ are workload assignment for trucks. As can be seen in Fig. 5, two trucks are arranged to work on six scheduling jobs. And Truck 1 is responsible for four jobs while Truck 2 is assigned two jobs. We assume that the first three jobs are discharging jobs and the second three jobs are loading jobs. The scheduling solution is that Truck 1 will handle jobs 6, 2, 4, 3, sequentially, while Truck 2 is assigned to handle jobs 5, 1, sequentially. According to the location solution, the first discharging job is located in ζ_2, the second discharging job is located in ζ_1 and the last discharging job is located in ζ_3, as shown in Fig. 6.

5 Computational Experiments

The computational experiments used to evaluate the performance of PSOs are discussed in this section. Four test problems of scheduling trucks are solved by Matlab R2001b running on a PC with Intel Core i5 2.20 GHz and 4 GB RAM.

The number of jobs (n) ranges from 8 to 300 while the number of trucks (m) ranges from 3 to 50 [17]. The four typical different scales of n and m are listed in Table 3. The initial location of trucks and pick-up/drop-off location are created following a uniform distribution in the two-dimension square from 0 * 0 m^2 to 1500 * 1500 m^2. And the earliest possible time of the jobs is randomly generated

Table 3 Four representative combinations of n and m

Instance	n	m
1	8	3
2	40	15
3	160	40
4	300	50

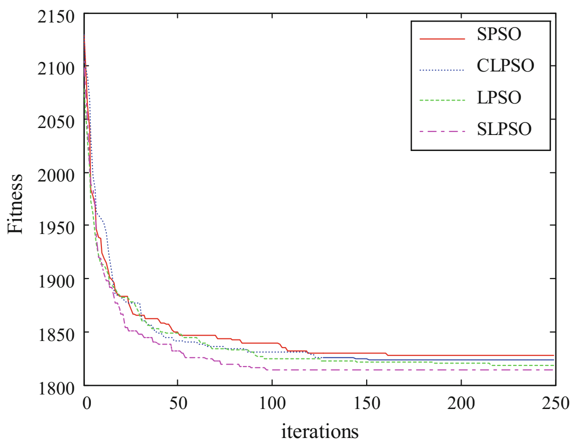
Table 4 Performance of the four PSOs on the test problem

Instance	(n, m)	Algorithm	Max	Min	Mean	Time(s)
1	(3,8)	SPSO	1.9466e+003	1.8140e+003	1.8279e+003	0.70
		CLPSO	1.8582e+003	1.8140e+003	1.8239e+003	1.30
		LPSO	1.8583e+003	1.8140e+003	1.8184e+003	0.17
		SLPSO	1.8147e+003	1.8140e+003	**1.8140e+003**	0.30
2	(15,40)	SPSO	2.7748e+003	2.3547e+003	2.5558e+003	13.26
		CLPSO	2.8587e+003	2.4160e+003	2.6460e+003	10.44
		LPSO	2.7941e+003	2.3125e+003	2.4494e+003	11.28
		SLPSO	2.5863e+003	2.2678e+003	**2.3932e+003**	9.30
3	(40,160)	SPSO	5.2197e+003	5.1623e+003	5.1710e+003	37.89
		CLPSO	5.3109e+003	5.2450e+003	5.2606e+003	39.53
		LPSO	4.9324e+003	4.6359e+003	4.7829e+003	38.08
		SLPSO	4.7277e+003	4.5305e+003	**4.3096e+003**	36.08
4	(50,300)	SPSO	6.9758e+003	6.7766e+003	6.8016e+003	49.30
		CLPSO	7.5080e+003	7.2925e+003	7.3626e+003	61.73
		LPSO	7.3979e+003	7.1155e+003	7.1690e+003	59.65
		SLPSO	6.3277e+003	6.1905e+003	**6.2306e+003**	48.08

following a uniform distribution of $\cup(0, 1500)$ (unit: second) and the due time of jobs is generated following a uniform distribution of $\cup(200, 500)$ (unit: second) [16]. The travel speed of trucks is 11.11 m/s and the two weight α_1 and α_2 are set to 0.6 and 0.4 as described in literature [8]. Each experiment is run twenty times.

It can be seen from the results presented in Table 4 that, with the increase of the number of jobs and trucks, the computational time grows rapidly. For the small-scale instance (i.e. $n = 3, m = 8$), all of the PSOs can find the minimum schedules for the problem. Figure 7 shows the average convergence rate on small-scale instance.

Fig. 7 The average evolution curve for small-scale instance ($n = 3, m = 8$)



Experimental study shows that the SLPSO outperforms the other three PSOs on all test problems. Our comparative results show that SLPSO performs well on small-scale problems and is promising for solving large-scale problems as well. It may be attributed to the fact that social learning mechanisms have the advantage of allowing individuals to learn behaviors from others without incurring the costs of individual trial-and-errors.

6 Conclusions

In this study, we addressed the cooperative scheduling strategy to reduce the unloaded rate and thus to cut back the make span. And PSO and its variants are applied to find optimal schedule strategy for the problem. A comprehensive set of test problems are used to compare the performance of the PSOs. The computational results demonstrated that the SLPSO performs better than all three PSOs on small-scale problems and large-scale problems as well. In future research, the development of more practical application will still be an emphasis on the study of integrated optimization model in container terminal operations.

Acknowledgments This work is partially supported by the National Natural Science Foundation of China (Grants nos. 71571120, 71271140, 71461027, 71471158 and 71501132), the Natural Science Foundation of Guangdong Province (Grant nos. 2016A030310074, 2016A030310067) and the Innovation and Development Fund Project of Shenzhen University (Grant no. 16XSCX04).

References

1. Nishimura, E., Imai, A., Papadimitriou, S.: Yard Trailer Routing at a Maritime Container Terminal. Transportation Research Part E Logistics & Transportation Review. 41(1):53–76 (2005)
2. Shang, J.: A heuristic Algorithm for the Integrated Yard Truck Scheduling in Container Terminal with Twin 40-Foot Quay Crane. Computer, Mechatronics, Control and Electronic Engineering (CMCE). 386–389 (2010)
3. Tsai, F.M., Chang, Y.M.: Optimization of Yard Truck Scheduling and Routing Model with Time-Space Network. Sustainable Development in Shipping and Transport Logistics. (2014)
4. Wang, Z.X., Chan, F.T.S., Chung, S.H., Niu, B.: Yard Truck Configurations for Efficient Operation of Container Terminals in Hong Kong. International Conference on Information Science, Electronics and Electrical Engineering. (2014)
5. Zheng, X., Shu, F., Mi, W.: Research on Genetic Algorithm-Based Simulation of Dynamic Container Truck Scheduling. IEEE International Conference on Computer Science and Automation Engineering. 131–135 (2012)
6. Wang, Z.X., Chan, F.T.S., Chung, S.H., Niu, B.: A decision support method for internal truck employment. Industrial Management & Data Systems. 114(9):1378–1395 (2014)
7. Ng, W.C., Mak, K.L., Zhang, Y.X.: Scheduling Trucks in Container Terminals Using a Genetic Algorithm. Engineering Optimization. 39(1):33–47 (2007)

8. Lee, D.H., Cao, J.X., Shi, Q., Chen, J.H.: A Heuristic Algorithm for Yard Truck Scheduling and Storage Allocation Problems. Transportation Research Part E Logistics & Transportation Review. 45(5): 810–820 (2009)
9. Lee, L.H., Chew, E.P., Tan, K.C., Wang, Y.: Vehicle Dispatching Algorithms for Container Transshipment Hubs. Operations Research-Spektrum. 32(3): 663–685 (2010)
10. Chung, S.H., Chan, F.T.S., Ip, W.H.: Minimization of Order Tardiness Through Collaboration Strategy in Multifactory Production System. IEEE Systems Journal. 5(1): 40–49 (2011)
11. Niu, B., Xie, T., Bi, Y., Liu, J.: Bacterial Colony Optimization for Integrated Yard Truck Scheduling and Storage Allocation Problem. IEEE Congress on Evolutionary Computation. 431–437 (2014)
12. Kennedy, J., Eberhart, R.: Particle swarm optimization. IEEE International Conference on Neural Networks, 1995. Proceedings. 1942–1948 (1995)
13. Liang, J.J., Qin, A.K., Suganthan, P.N., Baskar, S.: Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions. Evolutionary Computation IEEE Transactions on. 10(3):281–295 (2006)
14. Kennedy, J., Mendes, R.: Population Structure and Particle Swarm Performance. Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on. 1671–1676 (2002)
15. Cheng, R., Jin, Y.: A Social Learning Particle Swarm Optimization Algorithm for Scalable Optimization. Information Sciences. 291(6):43–60 (2015)
16. Wang, Z.X., Chan, F.T.S., Chung, S.H., Niu, B.: Minimization of Delay and Travel Time of Yard Trucks in Container Terminals Using an Improved GA with Guidance Search. Mathematical Problems in Engineering. 2015:1–12 (2015)
17. Ng, W.C., Mak, K.L., Zhang, Y.X.: Scheduling Trucks in Container Terminals Using a Genetic Algorithm. Engineering Optimization. 39(1):33–47 (2007)
18. Tasgetiren, M.F., Sevkli, M., Liang, Y.C., Gencyilmaz, G.: Particle Swarm Optimization Algorithm for Permutation Flowshop Sequencing Problem. Ant Colony Optimization and Swarm Intelligence, International Workshop. 382–389 (2004)
19. Shi, Y., Eberhart, R.C.: Parameter Selection in Particle Swarm Optimization. International Conference on Evolutionary Programming Vii. Springer-Verlag. 591–600 (1998)
20. Niu, B., Xie, T., Tan, L.J., Bi, Y., Wang, Z.X.: Swarm Intelligence Algorithms for Yard Truck Scheduling and Storage Allocation Problems. Neurocomputing. 188: 284–293 (2015)