Optimization Reshuffling Algorithm under Multiple Constraints for Container Terminal Yard Operations

Wei JIANG, QingPing GAO

College of Engineering, ZheJiang Normal University, Jinhua, 321004, China

Abstract — In order to improve the solution accuracy and efficiency of container reshuffling optimization algorithm for container wharf stacking under constrain, we propose to use chaos entropy disturbance variation and differential evaluation (DE) algorithm. Firstly, i) we develop a problem model for container stacking reshuffling process, ii) we present its optimized objective function and constraint condition. Secondly, i) we use differential evolution algorithm, ii) apply chaos entropy to improve population initialization process of differential evolution algorithm, iii) we improve individual distribution of the initial population, iv) we design a disturbance variation method to increase the diverse hold facility of individuals, v) which will improve the optimized accuracy and efficiency. Finally, we verify the effectiveness of our method through experimental comparisons.

Keywords - chaos entropy; disturbance variation; differential evolution algorithm; container; constraint of multi-element; container reshuffling optimization

I. INTRODUCTION

Stacking is critical area for production organization of container wharf. Its resources mainly include hoisting equipment and stacking slot space. Dispatch and management to these resources directly affects overall operational efficiency of container wharf.


The paper mainly aims at container reshuffling problem model of container stacking to perform research and presents its optimized objective function and constraint condition, and performs model optimization by adopting differential evolution algorithm. At the same time, to improve performance of differential evolution algorithm, make use of chaos entropy initialization to perform population initialization and design disturbance variation way, and realize performance improvement of differential evolution algorithm. Finally, verify effectiveness of mentioned method through experiment simulation.

II. PROBLEM MODEL OF CONTAINER STACKING

CONTAINER RESHUFFLING

A. Problem Description

The key of BRP is to affirm optimum location of falling container, to minimize reshuffling quantity. However, when operation process of actual wharf transtainer is considered, it is found that reshuffling quantity does not comprehensively reflect index of port waiting time of ship. Supposed that container reshuffling only appears in the same bay, container reshuffling process of exit container includes movement of car and hoisting equipment on transtainer. Although compared with a container reshuffling time, movement time of car and hoisting equipment is shorter, wharf is always in continuous operation state, and accumulation formed by scale economy cannot be ignored. Under the scheme where the least reshuffling quantity appears, it will be further pursued that movement amount of car and hoisting equipment on transtainer can be the smallest, which will further shortens time of overall container reshuffling operation.

Respectively remarked state of container $i$ before container reshuffling and after container falling as $(s_i,t_i)$ and
\((s', t')\), and of which \(s\) represents stack location, and \(t\) represents layer location. Therefore container reshuffling operation cost of transtainer to container \(i\) mainly includes \(C_c\) unit of car movement and \(C_s\) unit of hoisting equipment, which is mentioned in this paper, and they can be expressed as:

\[
C_c = |s' - s_i| \quad (1)
\]

\[
C_s = (S + h) - t_i + |t' - (S + h)| = 2(S + h) - t_i - t'_i \quad (2)
\]

In the formula, \(S\) represents maximum allowable stockpiling height of stacking, and \(h\) represents safety height during operation process (\(h\) units of layer direction and \(h \geq 1\)). During operation process, hoisting equipment has to lift container to safety height and moves through car, i.e. unit amount of distance between container and the greatest stockpiling height. Formula (1) represents movement amount of moving direction (direction \(y\)) of car on transtainer and formula (2) represents movement amount of vertical direction (direction \(z\)) of hoisting equipment on transtainer, which is as shown in Fig.1.

**B. Fundamental Assumption**

Assumption 1: stockpiling state of exit container in stacking bay and shipment order are known. In actual operation, before shipment, container exit order of each container has been decided on the wharf.

Assumption 2: container lifting and container overturn appear in the same bay, and that is to say that movement of large vehicle on transtainer is not considered. Because movement of large vehicle on transtainer is relatively time-consuming, in ship loading and unloading process of wharf, frequent switch of transtainer among different bays shall be avoided, and it shall be the phenomenon that after a bay is finished, it moves to another bay.

Assumption 3: only perform container overturn to blocking container of objective container, and that is to say that when objective container is lifted, principle of last-in first-out shall be followed. Only perform container overturn operation to container above objective container of the same stack.

Assumption 4: unit cost of movement of car on transtainer in direction of stacking column is equal to that of movement of hoisting equipment in the direction of layer.

**C. Mathematical Model**

Objective function of container overturn problem of container stacking can be defined as:

\[
\max TC = W_c \times \sum_{m=1}^{M} V_{mn} + W_s \times \sum_{m=1}^{M} (C_c + C_s) \times X_{m, \max} \quad (3)
\]

In the formula, \(TC\) is container overturn cost, including two parts of the least container overturn times and the least movement amount of transtainer. \(W_c\) and \(W_s\) respectively represents container overturn times and weight coefficient of movement amount of transtainer and they are constant value set. Considering that overturn times is main influence factor, set \(W_c = 100, W_s = 1\) and \(C_c\) and \(C_s\) respectively represents movement amount of hoisting equipment and car on transtainer.

**C1. Constraint condition of container overturn problem of container stacking is:**

\[
b_{\max} = b_{\max(n-1)} + \sum_{r \neq i}^{} X_{y, m(n-1)} - \sum_{r \neq i}^{} X_{y, m(n-1)} - Y_{m(n-1)} \quad \forall r, t, n \quad (4)
\]

In the formula, \(m = 2, \cdots, M\). Operation to some location includes container overturn and container lifting, and container overturn can be container falling location, and it can also be blocking container location. Therefore, the second item and the third item on the right of equal sign of formula (8) represents effect of container overturn on location variable \(b_{\max}\), and the fourth item represents effect of lifting container (to the ship) on \(b_{\max}\).

\[
V_{mn} = V_{m(n-1)} + \sum_{r \neq i}^{} Y_{m(n-1)} \quad \forall n \quad (5)
\]

In the formula, \(m = 2, \cdots, M\). Formula (5) represents constraint of location variable when container is on the ship. At the \(m\) th operation, value of \(V_{mn}\) is decided by previous operation and variable \(Y_{m(n-1)}\) which represents whether there is container transported to the ship.

\[
\sum_{r \neq i}^{} b_{\max} + V_{mn} = 1 \quad \forall m, n \quad (6)
\]

Formula (6) ensures that every container has a specific location, either on the ship or in the stacking.

\[
\sum_{r \neq i}^{} b_{\max} \leq 1 \quad \forall r, t, m \quad (7)
\]
WEI JIANG et al: OPTIMIZATION RESHUFFLING ALGORITHM UNDER MULTIPLE CONSTRAINTS FOR …

Formula (7) represents there can only be a container to the largest extent in any container location of stacking.

\[
-\sum_{s} b_{ms} + \sum_{s} b_{mn,1} \leq 0, \forall m, n, r, t = 1, \cdots, H - 1
\]  

(8)

Formula (8) ensures that no container at every column of stacking is in suspended state.

\[
\sum_{r,i,n} b_{mn} + \sum_{r,i,n} Y_{mn} \leq 1, \forall m
\]  

(9)

Formula (9) represents every operation can only be container lifting or container overturn, and that is to say that every operation is only aimed at one container.

\[
\sum_{Y_{ms}} \geq \sum_{Y_{(s+1)s}} + 1, \forall n = 1, \cdots, N - 1
\]  

(10)

Formula (10) guarantees that container lifting of container is performed according to established container lifting order, i.e. 1, 2, . . . .

\[
\left\{ \begin{array}{l}
b_{ms} \in \{0, 1\}, \forall r, t, n, m \\
X_{mn} \in \{0, 1\}, \forall r, t, i, j, n, m \\
Y_{mn} \in \{0, 1\}, \forall r, t, n, m \\
V_{mn} \in \{0, 1\}, \forall n, m \\
\end{array} \right.
\]  

(11)

Constraint (15)-(18) guarantees that above decision variables are all 0-1 variable.

III. CHAOS HEURISTIC DIFFERENTIAL EVOLUTION ALGORITHM

A. Basic DE Algorithm

DE is an algorithm based on population evolution, having feature of optimal solution of memory individual and information sharing within population, and that is to say that it optimizes solution of problem through cooperation and competition among individuals within population[8].

Firstly, a group of random initialized population has to be gained in the algorithm:

\[
X^{0} = \left[ x_{1}^{0}, x_{2}^{0}, \cdots, x_{N_{p}}^{0} \right]
\]

\(N_{p}\) is population scale. After a series of specified operations, the \(s\) th generation individual is evolved to:

\[
x_{i}^{s} = \left[ x_{i1}^{s}, x_{i2}^{s}, \cdots, x_{iD}^{s} \right]
\]

In the formula, \(D\) is dimension[9] of optimization problem.

DE/rand/1/bin and DE/best/2/bin: Two kinds of basic variation ways are listed here, i.e. DE/rand/1/bin and DE/best/2/bin.

\[
x_{m} = x_{t}^{s} + F \left( x_{i1}^{s} - x_{i2}^{s} \right)
\]  

(12)

In the formula (1) and (2), \(x_{i1}^{s}, x_{i2}^{s}, x_{i3}^{s}, x_{i4}^{s}\) are random individuals that are different to each other; \(x_{i1}^{s}\) is individual that has the greatest fitness among present population; \(F \in [0, 2]\) is zoom factor.

Interlaced strategy is: supposed that interlace operation is performed to individual \(x_{i}^{s}\) and \(x_{m}^{s}\) among population and test individual \(x_{T}^{s}\) is generated, to guarantee evolution of individual, make \(x_{T}^{s}\) at least have a bit contributed by \(x_{m}^{s}\) through random selection. As for other bits, crossover probability factor \(CR\) will be used and interlace operation equation is:

\[
x_{ij} = \left\{ \begin{array}{ll}
x_{mj} & \text{rand} \leq CR \\
x_{ij} & \text{rand} > CR
\end{array} \right.
\]  

(14)

As for selecting operation, “greedy” search strategy will be adopted, and those with higher adaptive value will be chosen as offspring.

\[
x_{i}^{s+1} = \left\{ \begin{array}{ll}
x_{i}^{s} & f\left(x_{i}^{s}\right) < f\left(x_{j}^{s}\right) \\
x_{j}^{s} & f\left(x_{j}^{s}\right) \geq f\left(x_{i}^{s}\right)
\end{array} \right.
\]  

(15)

Repeat above operation and end until offspring that meets adaptive value condition is generated [10–12].

B. Chaos Entropy Initialization

There are many rules to generate chaos, and what is adopted here is the most classic logistic model.

\[
cx_{i}^{k+1} = u \cdot cx_{i}^{k} \cdot \left( 1 - cx_{i}^{k} \right), i = 1, 2, ..., n
\]  

(16)

In the formula, \(cx_{i}^{k}\) is value of \(cx_{i}\) after the \(k\) th chaos evolution. When following conditions are met: \(u = 4\) , \(cx_{i} \in [0, 1]\) and \([0.25, 0.5, 0.75]\) , chaos phenomenon will be generated, and \(cx_{i}\) traverses within \([0, 1]\), which is as shown in Fig.1.

When value of variable is divided into \(x_{i} \in [a, b] \neq [0, 1]\), switch can generally be performed through following formula[16]:

\[
 cx_{i} = \left( x_{i} - a_{i} \right) / \left( b_{i} - a_{i} \right)
\]  

(17)
\[ x_i = a_i + cx_i(b_i - a_i) \]  

C. Disturbance Variation

An improved variation way is presented here:

\[ x_{i}^{r+1} = x_{i}^{r} + F \left( x_{i}^{r} - x_{i}^{r} + x_{i}^{r} - x_{i}^{r} \right) \]  

In above variation way, as for the \(i\) th individual \(x_{i}^{r}\) of the \(t\) th population, \(x_{i}^{r}\) is still used as base vector, and it represents that variation is performed on the basis of \(x_{i}^{r}\), which is beneficial to keep diversity of initial population while evolution direction of population is considered. The following item \(x_{i}^{r} - x_{i}^{r}\) is brought in as disturbance, which can keep great difference of individual after variation as they can, thus keeping diversity of population.

As for selection of \(r1, r2, r3\), we shall consider evolution direction of population while we perform random selection. We shall take following selection way. Generate 3 unequal numbers at random, \(c1 \neq c2 \neq c3 \in Z[0,1]\), and select according to size of its individual objective function value. As for minimization problem, select the minimum objective function value in \(r1\); and select the maximum objective function value in \(r3\), i.e.

1. \(r1 = \text{find} \{c1, c2, c3\}, \quad \text{s.t. min} \{\text{val}(x_{i}^{r}), \text{val}(x_{i}^{r}), \text{val}(x_{i}^{r})\}\)
2. \(r2 = \text{find} \{c1, c2, c3\}, \quad \text{s.t. mid} \{\text{val}(x_{i}^{r}), \text{val}(x_{i}^{r}), \text{val}(x_{i}^{r})\}\)
3. \(r3 = \text{find} \{c1, c2, c3\}, \quad \text{s.t. max} \{\text{val}(x_{i}^{r}), \text{val}(x_{i}^{r}), \text{val}(x_{i}^{r})\}\)

Processing in this way can make individual after variation be its own basic variable, which keeps great difference as it can be, thus keeping diversity of population. At the same time, select \(r1, r2, r3\) in order way, which considers evolution direction of population, so it is an efficient variation way.

D. CHDE Algorithm

Algorithm steps:

Step1: Set population size as \(NP\), dimension as \(D\), zoom factor as \(F\), crossover probability factor as \(CR\), and the maximum iteration algebra as \(G\) and searching space as \([t^0, u^0]\) and \(t = 1\).

Step2: Within \([t^0, u^0]\), initialize population \(P_1\) with size of \(NP\) by using method based on average entropy and calculate its fitness.

Step3: As for the \(t+1\) generation population \(P_{t+1}\), perform variation operation with (11) formula to generate...
new population $P_t$. Calculate fitness $f(x'_i)$ of each individual $x'_i$ in $P_t$.

Step4: Map vector of individual average value of present population as chaos variable $cx'_i \in [0,1]$.

Step5: New location point $cv'_i$ will be gained after single-step chaotic motion of $cx'_i$ by formula (5), and reflect $x'_i$ to common variable $y'_i$ within $[a_i, b_i]$ after conversion of formula (6), and calculate its fitness $f(y'_i)$.

Step6: Compare fitness $f(x'_i)$, $f(y'_i)$ and individual fitness $f(x_{i-1}^*)$ of last generation. Adopt greedy selection way and select optimal individual as individual of the next generation population.

Step7: Judge if terminal condition is met, and if it is met, then end iteration, otherwise turn to Step3, $t = t + 1$ .

IV. EXPERIMENTAL ANALYSIS

Combined with examples presented in Fig 1, computational process of FBS algorithm is as shown in Table 1. The table presents location of each objective container before container reshuffling and after container reshuffling and 3 kinds of evaluation indexes of every container reshuffling operation. Seen from it, with progressing of container reshuffling operation, basic container reshuffling quantity of stacking container tends to be small, but cost of transtainer and container reshuffling quantity increase with it.

To further verify effectiveness of algorithm, compare results in this paper with results in literature [6]. Design 2 groups of computational examples according to size of bay. Those lower than the 5th column and the 5th layer are small-scale computational example and another group is large-scale computational example. The comparison results are as shown in Table 2. In the table, the first column is problem scale, respectively representing column and layer height of bay. The second column is basic container reshuffling quantity of problem. Generate 10 groups of computational examples at random and calculate its average value to every problem scale. The third column is solution result of heuristic algorithm of Kim KH (referred to as KH algorithm), including container reshuffling quantity and consumption time of algorithm. The fourth column is solution result of algorithm mentioned in this paper, of which RC represents cost of transtainer. Seen from time of algorithm, average consumption time of algorithm in this paper is lower than KH algorithm.

<table>
<thead>
<tr>
<th>Order No. of container</th>
<th>Before container reshuffling</th>
<th>After container reshuffling</th>
<th>BRN</th>
<th>RC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Column</td>
<td>Layer</td>
<td>Column</td>
<td>Layer</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bay $s \times t$</th>
<th>BRN</th>
<th>KH algorithm</th>
<th>Algorithm in this paper</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Container reshuffling quantity</td>
<td>Container reshuffling time</td>
</tr>
<tr>
<td>3\times3</td>
<td>2.4</td>
<td>3.8</td>
<td>0.1</td>
</tr>
<tr>
<td>3\times4</td>
<td>4.2</td>
<td>5.7</td>
<td>0.1</td>
</tr>
<tr>
<td>3\times5</td>
<td>5.8</td>
<td>8.7</td>
<td>0.14</td>
</tr>
<tr>
<td>3\times6</td>
<td>6.2</td>
<td>8.8</td>
<td>0.16</td>
</tr>
<tr>
<td>4\times4</td>
<td>6.8</td>
<td>13.4</td>
<td>0.16</td>
</tr>
</tbody>
</table>

Comparing and analyzing results of algorithm, we can conclude that with increasing of problem scale, basic container reshuffling quantity shows an increasing tendency, and container reshuffling quantity increases with it, and movement amount of transtainer increases in a certain way. What is presented in Fig 3 is comparison between two algorithms and basic container reshuffling quantity. Seen from the Fig 3, in terms of solution quality, algorithm in this paper is closer to basic container reshuffling quantity, and has better performance compared with literature [6]. At the same time, in terms of solution speed, compared with KH
algorithm and FBS algorithm, better quality is shown in it. Solution time is shortened by 44.4% on average.

V. CONCLUSION

This paper puts forward container reshuffling optimization method of container stacking under constraint of multielement on the basis of chaos entropy disturbance variation DE algorithm, brings in differential evolution algorithm, and performs optimization study to problem model of container stacking container reshuffling. It makes use of chaos entropy to improve population initialization process of differential evolution algorithm, improving optimization accuracy and efficiency and verifies effectiveness of mentioned method through experiment comparison. Verify above algorithm under laboratory condition. But under practical condition, performance of algorithm remains to be further researched and verified, which will be emphasis of next research.

ACKNOWLEDGEMENT

The Zhejiang Provincial Natural Science Foundation of China under Grant No. LQ13G020010, and The key subject supported by Zhejiang Federation of Humanities and Social Sciences Circles of China under Grant No. 2014Z074.

REFERENCES