

Intelligent ATM replenishment optimization based on hybrid genetic algorithm

Jiecong Tang*, Shipeng Wang[†], Tao Bai[‡], Songfeng Lu[§], Jing Xiong[§]

* Software Development Center of Industrial and Commercial Bank of China Limited, Zhuhai, China

[†]School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China

[‡]School of Cyberspace Security, Huazhong University of Science and Technology, Wuhan, China

[§]School of Cyberspace Security, Huazhong University of Science and Technology, Wuhan, China

The corresponding author's e-mail is d201980943@hust.edu.cn or 358500196@qq.com

Abstract—To solve the ATM replenishment optimization problem, we conducted in-depth research and analysis on it, established a counterfeit model, and proposed an improved algorithm to solve it. Combine the global search capability of Genetic Algorithm (GA), the excellent convergence capability of Particle Swarm Optimization (PSO) and the local search capability of Simulated Annealing (SA) to propose an improved hybrid algorithm frame. Combine the actual production case of a bank's ATM replenishment planning to verify the effectiveness of the algorithm, the experimental results show that the proposed algorithm is effective in solving ATM replenishment optimization problem, and the proposed mathematical model is useful for the analysis of ATM replenishment optimization problem.

Index Terms—ATM replenishment optimization, improved genetic particle swarm optimization algorithm, ATM replenishment modeling, genetic algorithm, particle swarm optimization

I. INTRODUCTION

In the banking industry, cash management refers to the overall management of the cash under the jurisdiction of the bank, and the rational distribution of the cash share of each bank branch and each banknote addition device, so as to maximize the efficiency of bank cash use and bank revenue. With the "Outline of the National Informatization Development Strategy" and the "Thirteenth Five-Year Plan of the National Informatization Plan" and other documents that promote digital transformation, banks are trying to integrate current financial technology methods to carry out digital reforms and intelligent transformations. Among them, cash Intelligent management is an important part of transformation. ATM cash replenishment planning is an important link in cash management. It mainly includes consideration of the limitation of ATM service quality, formulating a cash distribution plan, solving which ATM machines need to be repaid and how much, as well as replenishing route planning and other issues. At present, it mainly relies on the professional experience of the practitioners, and there are problems such as low work efficiency, poor accuracy of banknote addition planning, high operating costs, and non-refined management.

Because the ATM cash demand in the banknote addition planning problem is dynamic and random, the solution to this problem is often divided into two steps, cash forecast and banknote distribution planning. Banknote distribution planning is based on the results of cash forecasts to give a banknote

addition plan that meets the minimum cost under the service quality. The accuracy of cash prediction is very important for banknote distribution planning. The methods to improve the accuracy include grouping ATMs into clusters and predicting by clusters [1], [2], introducing fuzzy clustering ideas to neural network models [3], and applying fuzzy clustering to the time series state space. Chaos modeling [4] and so on. This research studies the optimization problem of single-vehicle banknote distribution planning under the condition of known cash forecast value and considering certain ATM service quality. This problem is composed of the problem of minimizing the cost of banknotes and the problem of bicycle traveling salesman. By formulating a scientific and accurate banknote adding schedule, under the premise of ensuring the quality of ATM service, minimize the cost of labor and vehicles in the process of banknote adding planning. Improve business efficiency and empower intelligent features. For small-scale banknote distribution planning problems, precise algorithms can be used to traverse the solution space to solve them, such as dynamic programming [5], pruning algorithm based on local search [6] and so on. However, for large-scale banknote distribution planning problems, the increase in dimensionality will cause the problem to fall into dimensional disasters, and the solution efficiency and accuracy will be greatly reduced.

This research proposes an improved genetic particle swarm simulated annealing algorithm (IGAPSOSA) based on genetic algorithm (GA) to solve the banknote distribution planning problem, and uses real data from a bank to verify the algorithm, in order to improve the efficiency of ATM banknote addition management and cash The refinement of management provides a solution.

II. ANALYSIS AND MODELING OF BANKNOTE ADDITION PLANNING

A. Layout analysis

The location distribution diagram of ATM machines is shown in Fig. 1, which shows the location diagram of 5 ATM machines. The cash refill truck departs from the bank cash center (Dep), passes through the ATM location where cash refill is needed, and replaces the old cash box of the ATM with the loaded cash box until it completes the cash refilling task in the cash refill plan of the day, and finally returns to the

cash refill. The banknote center, unload the old banknote boxes and count the storage. For all ATM machines, the number is incremented starting from 1. There is arc connection between ATM machines or between warehouse and ATM machine, which represents the transportation cost between two points. The cash box value of the ATM machine represented by P_i is represented by I_i . As the deposit and withdrawal behavior occurs, I_i changes accordingly. It is generally believed that deposits and withdrawals occur within the effective deposit and withdrawal time.

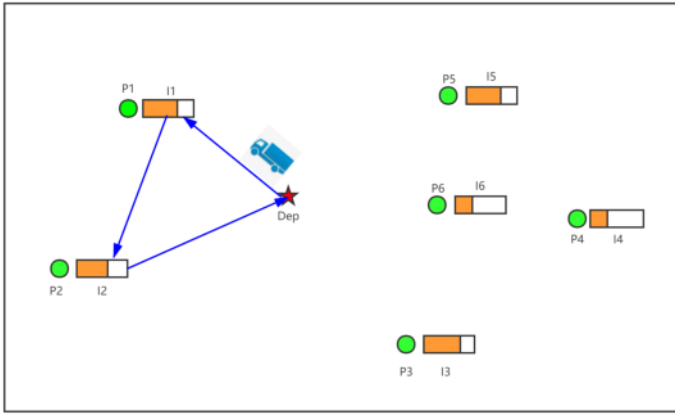


Fig. 1. ATM layout diagram.

The process of banknote addition planning is that the day before, the bank staff gave a banknote addition plan for the next day based on experience. The plan included which ATM machines to add banknotes, how much money was added to the corresponding ATM machines, and the order of banknote addition. The banknote refilling vehicle takes Dep as the starting point, traverses the corresponding ATM machine according to the banknote refilling schedule, and returns to Dep after completing the banknote refill. Different order of adding money corresponds to different routing cost and different service quality. The optimization goal is to minimize the total routing cost under the premise of ensuring a certain service quality within a period T and meeting the bank's constraints. For example, the form of the banknote replenishment plan is as follows, $[(1, 30), (2, 50)]$, which means that the replenishing vehicle starts from the warehouse and first goes to the P1 machine to add 300,000 yuan, then to the P2 machine to add 500,000 yuan, and finally back to the warehouse center.

The quality of service is mainly measured by two factors: lack of money and full money. The behavior of lack of money means that the balance in the ATM machine does not reach the customer's withdrawal limit, which results in the inability to withdraw money; the behavior of full money means that the balance of the ATM machine reaches the limit of the ATM cash box, which results in the inability to deposit.

Since the lack of money and full money can occur at any time, it is not convenient for statistics. Here for each ATM machine P_i , based on the predicted withdrawal value O_p^d and

predicted deposit value In_p^d of the machine in the next 8 days, statistics are made at two time points, the time when the cash-adding vehicle reaches the point P_i and the one-day trip of the cash-adding vehicle ends. At the moment, if one of the two time points is short of money or full of money, it is deemed that P_i is short of money or full of money.

The bank's restrictions include mandatory clearing of machines within a period and requirements for cash utilization rate. Compulsory machine cleaning means that in order to reduce operation and maintenance costs, and to discover ATM machines that have failed in time, each ATM machine must add money at least once in a cycle. The cash utilization rate is an empirical indicator used by banks to measure the cash holdings of ATMs.

B. Modeling

1) *Hypothesis*: Due to the complexity of the problem, for the simplicity of the mathematical model, the following assumptions are made:

- Suppose that the daily deposit and withdrawal behavior of ATM machines occurs from 9 to 19:00, and the balance in the ATM cash box remains unchanged in other time periods.
- Assume that the cash consumption speed of ATM machines is uniform. c. The time consumed by the banknote transporter for banknote exchange is negligible.
- The money-adding car can add money at most once a day.
- An ATM machine can be accessed at most once in a day.
- The state of lack of banknotes and full banknotes is judged by the state of the cash box at the time when the banknote refilling vehicle arrives at the ATM and when the refilling vehicle ends its day trip.
- Compared with the capacity of the cash box, the capacity of the cash-added car is infinite.
- The cash center has unlimited banknotes, which can meet the needs of any ATM plus banknotes.
- The cash-adding car is running at a constant speed.

2) *Cost analysis*: Given that the ATM machine numbers set $P = \{1, 2, \dots, N\}$, $\{0\}$ represents the cash center, and P can be regarded as the set of nodes in all directed graphs. Any edge set composed of ATM and ATM or ATM and cash center $\{A(i, j) | i, j \in P \cup \{0\}\}$. Period $T = \{1, 2, \dots, M\}$. Considering the shortage rate, full currency rate, and cash usage rate, find the minimum routing cost in the period T . In order to ensure that there must be a solution in the end, we use the above three conditions to be added to the cost of the model, and the model has no dimensions. The costs included in the model are described below:

- a. The cost of money shortage. If the ATM is in the money adding plan, the penalty cost is calculated by judging whether the ATM is short of money at two time points, which are when the money adding vehicle arrives at the ATM and after the effective deposit and withdrawal time of the day; if the ATM is not adding money In the plan,

it is only judged whether there is a shortage of money after the effective deposit and withdrawal time on that day, and the penalty cost is calculated. The total shortage cost of all P sets in period T is

$$C_{lack} = \sum_{d \in T} \sum_{p \in P} l_p \Delta_p^d \quad (1)$$

if $r_p^d=0$,

$$\Delta_p^d = \begin{cases} 1, & I_p^{d-1} - Q_p^d < 0 \\ 0, & \text{other.} \end{cases} \quad (2)$$

else,

$$\Delta_p^d = \begin{cases} 1, & I_p^{d-1} - \frac{C_p^d}{N^d} Q_p^d < 0 \\ & \text{or } r_p^d - (1 - \frac{C_p^d}{N^d}) Q_p^d < 0, \\ 0, & \text{other.} \end{cases} \quad (3)$$

Where l_p is the currency shortage penalty coefficient; I_p^{d-1} is the remaining amount of node p at the end of day $d-1$, I_p^0 is the remaining amount of node p the day before the banknote increase plan, and the update formula for I_p^d is

$$I_p^d = \begin{cases} \max(0, I_p^{d-1} - \frac{C_p^d}{N^d} Q_p^d), & r_p^d = 0 \\ \max(0, r_p^d - (1 - \frac{C_p^d}{N^d}) Q_p^d), & r_p^d \neq 0. \end{cases} \quad (4)$$

In the above equations, C_p^d is the note adding sequence number of node p on day d . All the nodes that need to add note on day d are counted. After solving the TSP problem for these nodes, the note adding sequence of each node can be obtained. N^d is the sum of the sequence of node machines that need to add money on day d , and the calculation formula is

$$N^d = \sum_{s \in S} \sum_{p \in P} y_p^s k_s^d. \quad (5)$$

Among them, $S = \{1, \dots, 2^\tau - 1\}$ is the set of banknote addition plans, considering the mandatory clearing requirements, so the minimum value of S is 1. s is the money adding plan, $s \in S$, for example, when $s = 18$ and $T = 6$, first convert to the corresponding binary form and then reverse the order to get the banknote addition mode (Pat) corresponding to s , which is 010010, which means that a cycle is six days, and banknotes are added on the second and fifth days. y_p^s is a decision variable. If the cash increase plan corresponding to node p is s , y_p^s takes the value 1, otherwise the value is 0. Each node p selects a money adding plan s , according to the Pat corresponding to the money adding plan s , we can get whether the node p adds money $k_s^d \in \{0, 1\}$ on the d day, and count all nodes k_s^d on the d day to get N^d . $Q_p^d = Q_p^d - I_n^d$ is the net demand of node p on day

d , where r_p^d is the amount added to node d on day d , calculated as follows

$$r_p^d = w_p^d + I_p^{d-1} \quad (6)$$

In the formula, w_p^d is the net adding-cash value of node p on day d . The net adding-cash value is calculated according to the cash adding plan s corresponding to node p . First, convert s to the corresponding Pat. If the d position of the Pat sequence is 0, the net adding-cash value is 0, otherwise, look backwards from the d^{th} position until you encounter the next position e with a value of 1. The net cash adding value is the net cash adding value from day d to day $e-1$ The sum of demand.

$$\sum_{i=d}^{e-1} Q_p^i. \quad (7)$$

- b. The full currency cost. The calculation is similar to the lack of currency cost, the total full currency cost of all P sets in the period T is

$$C_{full} = \sum_{d \in T} \sum_{p \in P} u_p \Theta_p^d \quad (8)$$

if $r_p^d=0$,

$$\Theta_p^d = \begin{cases} 1, & I_p^{d-1} - Q_p^d > P_c^p \\ 0, & \text{other.} \end{cases} \quad (9)$$

else,

$$\Delta_p^d = \begin{cases} 1, & I_p^{d-1} - \frac{C_p^d}{N^d} Q_p^d > P_{c_p} \\ & \text{or } r_p^d - (1 - \frac{C_p^d}{N^d}) Q_p^d > P_{c_p}, \\ 0, & \text{other.} \end{cases} \quad (10)$$

Where P_{c_p} is the capacity of the cash box of node p , that is, the upper limit of the amount of money added; u_p is the penalty coefficient for full cash at node p .

- c. The cost of cash utilization rate. The cost of cash utilization rate is calculated according to the average cash utilization rate of node p in period T . The total cash utilization rate cost of all P collections in period T is

$$C_{circu} = \sum_{p \in P} t_p \Gamma_p \quad (11)$$

$$\Gamma_p = \begin{cases} 1, & \text{if } \overline{\chi_p} < \phi \\ 0, & \text{other.} \end{cases} \quad (12)$$

Among them, t_p is the cost coefficient of the cash utilization rate, ϕ is the lower bound of the cash utilization rate.

$$\overline{\chi_p} = \frac{\sum_{d \in T} \chi_p^d}{T}, \quad (13)$$

which is the average cash utilization rate of node p in period T , where χ_p^d is the cash utilization rate of node p on day d , and the calculation formula is

$$\chi_p^d = \begin{cases} \frac{O_p^d + v_p^d}{In_p^d + r_p^d}, & \text{Add money at } p \text{ on } d, \\ \frac{O_p^d}{In_p^{d-1} + r_p^d}, & \text{other.} \end{cases} \quad (14)$$

In the formula, v_p^d is the unloading value of node p on day d . Since the behavior of adding money is to use a new cash box in exchange for the old one, the unloading value here is the remaining amount of the old one, and the calculation formula is

$$v_p^d = \begin{cases} I_p^{d-1} - \frac{C_p^d}{N_p^d} Q_p^d, & \text{Add money at } p \text{ on } d, \\ 0, & \text{other.} \end{cases} \quad (15)$$

- d. The routing cost is calculated based on the mileage charged. The calculation formula of the road cost is

$$C_{route} = \sum_{d \in T} \sum_{(i,j) \in A} c_{ij} x_{ij}^d. \quad (16)$$

Where c_{ij} is the travel distance from point i to point j ; x_{ij}^d is a decision variable. If the banknote adding route passes (i, j) on the d day, the value is 1, otherwise it is 0.

- 3) *Modeling*: The optimization objective function is

$$\min(C_{lack} + C_{full} + C_{circu} + C_{route}), \quad (17)$$

and the constraints are

$$\sum_{s \in S} y_p^s = 1 (\forall p \in P), \quad (18)$$

$$\sum_{j \in P \cup \{0\}} x_{ij}^d = \sum_{s \in S} k_s^d y_p^s (\forall p \in P; \forall d \in T), \quad (19)$$

$$\sum_{j \in P \cup \{0\}} x_{ij}^d = \sum_{j \in P \cup \{0\}} x_{ji}^d (\forall p \in P \cup \{0\}; \forall d \in T), \quad (20)$$

$$\sum_{j \in P \cup \{0\}} x_{0j}^d \leq 1 (\forall d \in T), \quad (21)$$

$$\sum_{i,j=R} x_{ij}^d \leq |R| - 1 (\forall R \subseteq P; \forall d \in T), \quad (22)$$

Among them, the formula (18) restricts that each point corresponds to only one money adding scheme in the period. The formula (19) and formula (20) are the restriction conditions of the out-degree and in-degree in the graph composed of the addition points. Formula (21) limits the departure from the banknote replenishment center at most once a day, that is, the banknote replenishment is carried out at most once a day. Formula (22) is the sub-loop elimination constraint condition of the TSP problem.

III. ALGORITHM DESIGN

The GA algorithm has been applied to the problem of banknote addition planning [7], but for large-scale banknote addition planning problems, the GA algorithm requires a large number of populations to find the optimal solution, the number of iterations is high, and the calculation efficiency is low. The particle speed update in the PSO algorithm consists of three parts: the speed information of the previous generation of the particle, the difference between the current particle position and the best position of the current generation, the difference between the current particle position and the best position in the historical iteration, and the overall particle is towards the global best value. Approaching, so the algorithm has high flexibility and good convergence. Aiming at the shortcomings of the GA algorithm, we use the PSO algorithm to accelerate the efficiency of the algorithm and improve the convergence speed. At the same time, in the final stage of the algorithm, the local search algorithm simulated annealing is used to optimize the algorithm, improve the algorithm's jumping out of the optimal, and global optimization. ability.

Based on the above, an improved genetic particle swarm simulated annealing algorithm (IGAPSOSA) is proposed. The algorithm is divided into three stages. The first stage uses the GA algorithm to optimize the population so that the convergence direction of the algorithm is kept correct; the second stage uses the PSO algorithm to speed up the convergence speed of the algorithm; the third stage uses the improved genetic algorithm based on SA GASA, to enhance the global optimization capability of the algorithm. In GASA, for each generation in GA, the individual with the best fitness is selected and sent to SA for optimization. If the result of SA processing is better than the old value, it is updated.

A. Coding

Each ATM machine corresponds to a banknote adding plan s in the cycle, s corresponds to the banknote adding mode Pat . Pat is a sequence of 01 with a length of T , which reflects the information of whether the ATM adds banknotes every day during the period T . We use the sequence composed of ATM serial numbers of all ATM machines as the gene expression of individuals in the population.

B. Population initialization

For each ATM, the initial s value is randomly selected from the range of $[1, 2^T - 1]$.

C. Fitness function

The fitness function is used to evaluate the pros and cons of individuals. For the problem of banknote adding planning, the cost function is regarded as the fitness function. The smaller the value of the cost function, the better the individual.

D. Select action

The selection operator determines the choice of individual parents who participate in the construction of the next generation. The convergence performance of GA is affected

by the selection operator. As the tournament mechanism can increase the selection of excellent genes to the greatest extent, which can increase the exploration ability of the algorithm, the tournament operator with $K = 2$ is used in this paper. For a population P of size N , N tournaments will be held. For each tournament, K individuals will be randomly selected from the population P , and the best individual will be selected as the parent of the next generation after the fitness comparison.

E. Cross operation

The crossover operator is responsible for generating the offspring individuals containing the excellent genes of the parents based on the genes of the parents. Since the position and number of uniform crossover points are not fixed, it is conducive to the generation of individuals who have integrated the excellent genes of their parents to a greater extent. The uniform crossover operator is used. Traverse the gene sequences of two parent individuals, and for each position, determine whether the position should be crossed or not according to the crossover probability. For example, one parent individual is $\{1, 32, 47, 2, 9\}$, the other parent individual is $\{3, 5, 54, 33, 4\}$, and the offspring individuals of the crossover are $\{1, 5, 54, 2, 4\}$ and $\{3, 32, 47, 33, 9\}$, the positions where the crossover occurs are 2, 3, and 5.

F. Mutation operation

The mutation operator increases the diversity of genes from generation to generation, thereby increasing the exploration ability of the algorithm. For the offspring individuals obtained by the crossover operation, traverse the genetic sequence of the individual and convert the code value of each position into the corresponding binary Pat form. For each position in the Pat , it is flipped according to the probability, and 0 is turned into 1 after flipping, 1 becomes 0 after flipping. For example, the individual is $\{1, 32, 47, 2, 9\}$, the corresponding Pat sequence is $\{1000000, 0000010, 1111010, 0100000, 1001000\}$, for position 1 of the gene sequence, after mutation It may be $\{0110001\}$, which means that there has been a mutation at the positions 1, 2, 3, and 7 of Pat , and the corresponding code value becomes 70. The other positions of the gene sequence undergo the same operation.

G. PSO population initialization

The population processed by the GA algorithm in the first stage is used as the initial population in the PSO stage.

H. SA cooling rate

The SA algorithm starts from a sufficiently high temperature, and decreases the temperature during operation until the termination condition (such as termination temperature or iterative condition) is reached. The temperature drop varies with the number of iterations t , see the formula

$$T_{t+1} = \frac{T_0}{1+t}. \quad (23)$$

Where T_0 is the initial temperature of simulated annealing.

I. SA Neighbor Solution

For the simulated annealing stage, the purpose of obtaining the neighboring solution is to enable the algorithm to expand from the current solution, not only to a better solution, but also to a larger search area, so as to enhance the development ability of the algorithm. We propose a selection algorithm based on roulette. The algorithm generates neighbor solutions based on the information of the lack of banknote matrix lm and the full banknote matrix um . The process is described as follows:

- 1) Sum the lack of banknotes matrix lm and the full banknotes matrix um to get the sum matrix all_{mat} .
- 2) all_{mat} is summed by line, and counts the number of full banknotes of each ATM machine in the cycle.
- 3) According to 2), construct a roulette selection algorithm to select the ATM serial number p to be processed.
- 4) Randomly select a processing operator from the three constructed processing operators to act on ATM machine p to obtain a new neighbor solution and return.

Among them, the processing operator is described as:

- Left shift operator: circularly shift the note adding Pat of ATM machine p to the left by one position, as shown in Fig. 2(a), the sequence of 1010010 is circularly shifted to the left to obtain 0100101.
- Right shift operator: circularly shift the money adding Pat of ATM machine p to the right by one position, as shown in Fig. 1(b), after circularly shifting the sequence of 1010010 to the left, 0101001 is obtained.
- Flip operator: randomly select a certain position of the note adding Pat of ATM machine p to flip, as shown in Fig. 1(c), select the fourth position in the sequence of 1010010 and flip to get 1011010.

IV. EXAMPLES AND ANALYSIS

Take a bank as an example. The bank manages 72 ATM machines and uses the above-mentioned IGAPSOSA algorithm for banknote addition planning. The bank only adds banknotes on working days and does not add banknotes on non-working days, so it plans to add banknotes When considering the use of cash on weekends. The ATM distribution diagram of the bank is shown below. The operating environment is Intel(R) Xeon(R) CPU E5-2620 v2 @ 2.10GHz, and the development language is Python.

A. Parameter setting and experiment

The banknote adding cycle $T = 5$, the length of Pat is 7, and the positions corresponding to Saturdays and Sundays are always 0, which means no banknotes are added. The crossover probability $P_{cross1} = 1.0$ of the GA algorithm in the first stage, the mutation probability $P_{mutate1} = 0.05$. In the second stage, the initial inertia factor $w = 0.9$ of the PSO algorithm, the initial individual cognitive factor $C_{personal} = 2.0$, and the initial group social factor $C_{global} = 2.0$. In the third stage of GASA, the crossover probability $P_{cross2} = 0.05$, the mutation probability $P_{mutate2} = 0.01$; the initial temperature

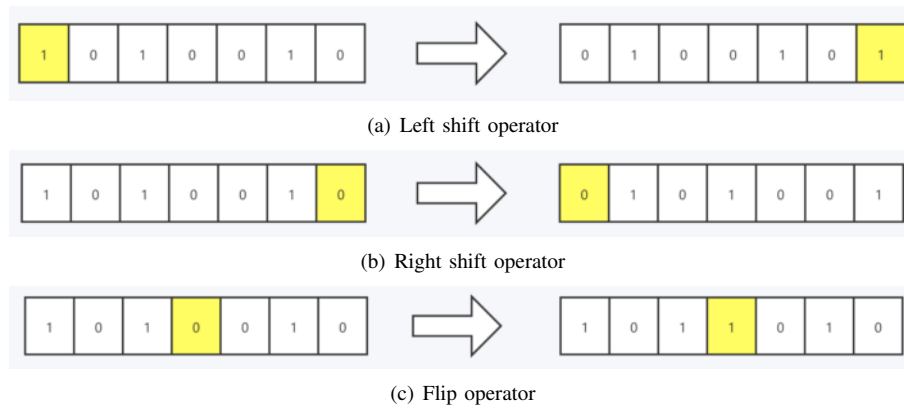


Fig. 2. Three operation operators defined (left shift, right shift, flip)

$T_0 = 70$ of SA, the end temperature $T_f = 0.001$, and the number of iterations for each temperature $L = 10$. The population size is set to 90, the number of iterations is 200, the first stage execution proportion $P_{GA} = 0.55$, the second stage execution proportion $P_{PSO} = 0.15$, and the third stage execution proportion $P_{GASA} = 0.30$. The calculation results are shown in Table I and Table II.

TABLE I
COMPARISON OF MAIN EVALUATION INDEX RESULTS OF EACH ALGORITHM

Algorithm	Total cost	Number of shortages	Number of full banknotes	Substandard number of Usage rates
IGAPSOSA	2948.465	13.5	0.5	4.25
GA	3087.431	14.9	0.5	4.1
PSO [8]	6355.833	57.4	0.4	51.3
GAPSO	3024.955	14.4	0.6	4.1
ABC*	4799.316	20.3	6.8	9.3

*The values in the ABC algorithm are calculated after rounding up, down, and rounding, and all the results are summarized and counted.

TABLE II
COMPARISON BETWEEN MANUAL AND IGAPSOSA BEST VALUE

	Cost	Number of shortages	Number of full banknotes	Substandard number of Usage rates
Artificial	3628.633	16	1	9
IGAPSOSA	2623.108	14	0	1

B. Result analysis

Table I is the average weekly data statistics of each algorithm in the data set. By comparing the total cost value, the IGAPSOSA algorithm is better than other algorithms, and the total cost of the GAPSO algorithm is reduced by 76.49. The number of shortages is also the best, which is better than GAPSO. The number of shortages is 0.9 times less. This is because the IGAPSOSA algorithm adds SA local search in the third stage compared with the GAPSO algorithm, and the SA hill climbing process plays a role, which is

convenient for jumping out of the local extreme value to find the global extreme value. The PSO algorithm has fewer times of adding money, although the routing cost is the lowest, but the phenomenon of lack of money and substandard usage is the most serious. Judging from the full banknote statistics of each algorithm, the phenomenon of full banknotes is relatively rare. The GA, IGAPSO, and IGAPSOSA algorithms are close to the number of times that the cash utilization rate is not up to standard. The analysis also shows that although the ABC algorithm has a good optimization ability in the test function, in actual problems, the convergence effect and the optimization ability are not as good as the IGAPSOSA, GAPSO and GA algorithms. This is because the ABC algorithm is suitable for solving continuous problems. For discrete problems, the rounding method cannot achieve good results.

Table II is a comparison between the best result of the IGAPSOSA algorithm and the result of manual decision in a selected period. The total cost value of the IGAPSOSA algorithm is 1005.522 less than the cost value of manual decision, the number of shortages is 2 times less, and the cash utilization rate is not up to the standard. The number is less than 8. It can be seen that there are more unqualified numbers of cash utilization rate in manual decision-making. According to the calculation method of cash utilization rate, cash utilization rate is calculated based on the average value within a week, and manual decision-making is based on each day. The optimal decision will result in an increase in the number of non-working days that are not up to standard. At the same time, according to the data, there are not a lot of cash shortages. This is mainly caused by bank G's failure to add cash on holidays. Withdrawals on holidays are more valuable. In some supermarkets and other places where there is a large flow of people, ATMs are used on holidays. The balance of the machine is easy to be withdrawn and the ATM machine is short of money. On the whole, the solution effect of the IGAPSOSA algorithm is better than manual decision-making, and it can be used to replace manual banknote addition planning.

V. CONCLUSION

Aiming at the bank cash management problem of banknote

addition planning, a mathematical model for analyzing the problem is established. Combining the advantages of GA, PSO and SA, the IGAPSOSA algorithm is proposed to solve the problem. Using real data from a bank to verify the effectiveness of the algorithm, the results show that the proposed algorithm reduces the total cost of GA by 4.5%, reduces the total cost of PSO by 53.6%, and reduces the total cost of GAPSO by 2.5%. The cost is reduced by 27.7%. As a result, the IGAPSOSA algorithm can be used to solve examples of banknote addition planning, which can meet the needs of banknote addition managers. The established banknote addition model has the usability to analyze actual banknote addition planning problems.

VI. ACKNOWLEDGEMENTS

Tao Bai would like to acknowledge financial support from the Fundamental Research Funds for the Central Universities, HUST:2020JYCXJJ078.

REFERENCES

- [1] Ekinci Y, Lu J, Duman E. Optimization of ATM cash replenishment with group-demand forecasts[J]. *Expert Systems with Applications*. 2015, 42(7): 3480-3490.
- [2] Jadwal P K, Jain S, Gupta U, et al. K-Means clustering with neural networks for ATM cash repository prediction[C]. Springer, 2017.
- [3] Seedig H G, Grothmann R, Runkler T A. Forecasting of clustered time series with recurrent neural networks and a fuzzy clustering scheme[C]. IEEE, 2009.
- [4] Vangala S, Vadlamani R. ATM Cash demand forecasting in an Indian Bank with chaos and deep learning[J]. *arXiv preprint arXiv:2008.10365*. 2020.
- [5] özer F, Toroslu I H, Karagoz P. Comparison of Integer Linear Programming and Dynamic Programming Approaches for ATM Cash Replenishment Optimization Problem[J]. *International Journal of Applied Metaheuristic Computing (IJAMC)*. 2020, 11(3): 120-132.
- [6] Larrain H, Coelho L C, Cataldo A. A variable MIP neighborhood descent algorithm for managing inventory and distribution of cash in automated teller machines[J]. *Computers & Operations Research*. 2017, 85: 22-31.
- [7] Kurdel P, Sebestyenova J. Modeling and optimization of ATM cash replenishment[C]. 2012.
- [8] Zhan Z, Zhang J, Li Y, et al. Adaptive particle swarm optimization[J]. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*. 2009, 39(6): 1362-1381.
- [9] M.R. Khouadjia, E.-G. Talbi, L. Jourdan, B. Sarasola, E. Alba, Multi-environmental cooperative parallel metaheuristics for solving dynamic optimization problems, *J. Supercomput.* 63 (3) (2013) 836–853.
- [10] T.J. Ai, V. Kachitvichyanukul, A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery, *Comput. Oper. Res.* 36 (5) (2009) 1693–1702, <https://doi.org/10.1016/j.cor.2008.04.003> arXiv:arXiv:1011.1669v3.
- [11] Ropke S, Pisinger D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science* 2006;40(4):455–72.
- [12] Goldberg AV. An efficient implementation of a scaling minimum-cost flow algorithm. *Journal of Algorithms* 1997;22(1):1–29.
- [13] Archetti C, Bertazzi L, Laporte G, Speranza MG. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science* 2007;41(3):382–91.
- [14] Archetti C, Bertazzi L, Hertz A, Speranza MG. A hybrid heuristic for an inventory routing problem, *INFORMS Journal on Computing*; 2011, doi: 10.1287/ijoc.1100.0439.
- [15] Bertazzi L, Paletta G, Speranza MG. Deterministic order-up-to level policies in an inventory routing problem. *Transportation Science* 2002;36(1):119–32.
- [16] Coelho LC, Cordeau J-F, Laporte G. The inventory-routing problem with transshipment, Technical Report 21, CIRRELT, Montreal, Canada; 2011.



Jiecong Tang received the Bachelor degree in Software Engineering and the Master degree in Computer Technology both from South China Agricultural University, in 2016 and 2018. His research interests include Artificial Intelligence, Deep Learning and Nature Language Processing.



Shipeng Wang received his Bachelor degree in Chemical Engineering and Technology in 2016 and his Master degree in Computer Application Technology in 2021, both from Huazhong University of Science and Technology, Wuhan, China. He is currently a Senior Engineer at Huawei Technologies Co., Ltd. in Shenzhen. His research interests are Internet of Things (IOT) and industry solutions.



Tao Bai received his Bachelor degree in Space Engineering from National Defence of Science & Technology in 2012, Changsha and his Master degree in mathematics from South-central University For Nationalities in 2018, Wuhan. He is currently a PhD student in Huazhong university of Science & Technology, Wuhan. His research interests include artificial intelligence, secure multiparty computing and blockchain.



Songfeng Lu was born in 1968. He received the Ph.D degree in Computer Software and Theory from Huazhong University of Science and Technology. He is a professor of Huazhong University of Science and Technology. His research interests include Artificial Intelligence, Quantum Computing and Information Security.



Jing Xiong received her Bachelor degree in network engineering from Xidian University in 2021. She is currently a master student in Huazhong University of Science and Technology, Wuhan. Her research interests include artificial intelligence, secure multiparty computing, and privacy protection.