

Analysis of AGV Optimal Path Problem in Smart Factory Based on Genetic Simulated Annealing Algorithm

Chaonan Fan^{1, a}, Shixin Li^{1, b, *}, Rong Guo¹, and Yuxin Wu²

¹Tianjin University of Technology and Education, Tianjin, China

²Institute of Microelectronics, Tianjin University, Tianjin, China

^afanchaonan0621@163.com, ^bli_shixin@sina.com

*corresponding author

Keywords: Genetic algorithm, Simulated annealing algorithm, AGV, MATLAB.

Abstract: This paper analyzes the optimal path problem of AGV for smart factories, and propose a hybrid algorithm combining genetic algorithm and simulated annealing algorithm. The objective function model is established according to the target optimization, the improved genetic algorithm and simulated annealing algorithm can be used to obtain better path and can effectively fall into local optimum. Simulation results show that this improved genetic simulated annealing algorithm is feasible in solving the AGV optimal path of smart factories.

1. Introduction

The production automation transformation of factory is the only way that promoted the manufacturing industry to industry 4.0. The intelligent factory constantly improves the level of enterprise automation production and reduces unnecessary labor costs, especially the material transportation in the production workshop. In the enterprise intelligent automation transformation, the most critical is AGV transportation system. AGV transportation system plays a certain role in improving working environment and production efficiency.

Genetic algorithm is a global probabilistic search algorithm based on the natural selection and evolution mechanism of biological world, but the standard genetic algorithm is not efficient in solving AGV path optimization problems. In this paper, the genetic algorithm in the intelligent algorithm is improved and combined with the simulated annealing algorithm to avoid the premature convergence of the genetic algorithm and the local optimal solution. This hybrid algorithm can strengthen its optimization ability in the application of AGV path.

2. AGV Path Problem Description for Smart Factory

AGV refers to a transport vehicle equipped with electromagnetic or optical guidance devices, which can travel along the specified guidance path, and has safety protection and various shifting functions. The AGV transportation system in the intelligent factory is completely independent of manual work. By issuing instructions through the computer and following certain position information for automatic walking and loading and unloading, its reliability and efficiency are much higher. The system can operate automatically and can adapt to all kinds of materials. We describe the AGV transport problem as: k different AGV to be transmitted at m mission points, each AGV needs to be sent to m mission points, each task point requires a different material delivered by the AGV. The transport time of each AGV and the order constraints of various materials in the task point are known. The objective of the problem is to ensure that each AGV's route in transmission will not interfere with each other and the transport distance can be minimized. The optimization goal here is to make AGV the shortest path in the transmission process.

The mathematical model is established by the above analysis: Make the task code $1, \dots, N$. The task $i(s_i, r_i)$ means the starting point of task i is s_i , the end is r_i , then the task $j(s_j, r_j)$ means the

starting point of task j is s_j , the end is r_j . So d_{sr} expression the shortest distance between task i and task j . x_{ij} expression whether to execute the task j after executing the task i . According to the problem description, the AGV optimization scheduling problem can be established as follows. The mathematical model is as shown in formula (1):

$$\min \sum_{i=0}^N \sum_{j=0}^N x_{ij} d_{sr} \quad (1)$$

s.t.

$$\sum_{i=0}^N x_{ij} = 1, j = 0, 1, \dots, N \quad (2)$$

$$\sum_{j=0}^N x_{ij} = 1, i = 0, 1, \dots, N \quad (3)$$

$$\sum_{i,j} x_{ij} = N \quad (4)$$

$$x_{ij} = \{0, 1\}, i, j = \{0, 1, \dots, N\} \quad (5)$$

Constraint condition(2)and(3)indicates that each task is only executed once. Constraint condition(4)indicates that a total of N tasks are executed. Constraint condition (5) indicates whether the road segment has passed.

3. Algorithm Design

3.1. Genetic Algorithm and Simulated Annealing Algorithm

Genetic algorithm (GA) is a stochastic parallel search algorithm based on the principles of natural selection and genetics. The application of genetic algorithm to solve the shortest path problem first needs to be encoded, then random initial population, calculate the adaptive function and selection rate, and then simulate the behaviors of inheritance, crossover, mutation, etc., to determine whether the next generation meets the demand. The above process is repeated until the convergence iteration condition is satisfied. Finally, when the number of iterations of the algorithm reaches the specified number of times, the solution with the highest fitness function is output, that is, the global optimal solution, and the function value of the solution is output, that is, the global optimal value. However, for genetic algorithms, the solution of the optimal path requires multiple iterations. The algorithm can quickly converge to the optimal extreme value by selecting the intersection, but in the later stage of optimization, since all the individual chromosomes in the population are very close, it is difficult to produce after the intersection.

The principle of simulated annealing algorithm is similar to the process of physical annealing: initialization temperature T_0 , initial solution V_0 , Random hopping within a certain range of the original solution produces a new solution V_1 , when $V_1 - V_0 > 0$, directly accept V_1 as a new solution. The final algorithm tends to the global optimal solution when the temperature tends to the lowest critical value, and it is not easy to fall into the local optimum.

3.2. Design of Genetic Simulated Annealing Algorithm(GSA)

First initialize the population: defining the population size as m , individual chromosome length is $n + 1$. The first n genes of the chromosome consist of n different values (representing different task points). The $n + 1$ gene is identical to the first gene, indicating that it returns to the original location (the first n values represent n different locations, and the order of the chromosomal genes is the order of the mission points passed by the AGV). The chromosome of the i individual in the initial population is U_{0i} , initial population optimal function value is D_0 . The specified initial temperature value is $T_0 = 100^\circ\text{C}$. T is the current temperature and T_{min} is the minimum temperature. Initialize the current iteration number is $g = 1$. Defining the fitness function:

$$f(D) = \begin{cases} (D_{max} - D) \left(\frac{T}{100} \right)^{\frac{1}{3}}, & D < D_{max} \\ 0, & D \geq D_{max} \end{cases} \quad (6)$$

D_{max} is the maximum function value of the individual that the algorithm can tolerate. The fitness formula can make the individual differences of the population become smaller as the temperature decreases. Therefore, it is possible to appropriately increase the fitness when the temperature is low, reduce the probability of the individual being selected, ensure the diversity of the individual population, and effectively improve the algorithm to jump out of the local optimum.

Next, update the chromosomes of the population individuals through selection, crossover, and mutation algorithms. Recalculate the local optimal function value D_{lob} and the local optimal solution U_{lob} of the new population, and compared with the previous global optimal function value D_{glb} . If $D_{lob} < D_{glb}$, then assign D_{lob} to D_{glb} . When $g < g_{max}$, the number of iterations is increased to continue to calculate the fitness. When $g = g_{max}$, reinitialize the population when T is greater than the T_{min} . When T is smaller than T_{min} , the temperature drop ends, and the global optimal function value D_{gob} and the global optimal solution U_{gob} are obtained. T_{min} is set to 0.1.

The flow chart of the intelligent factory AGV scheduling problem based on genetic simulated annealing algorithm is shown in Figure 1.

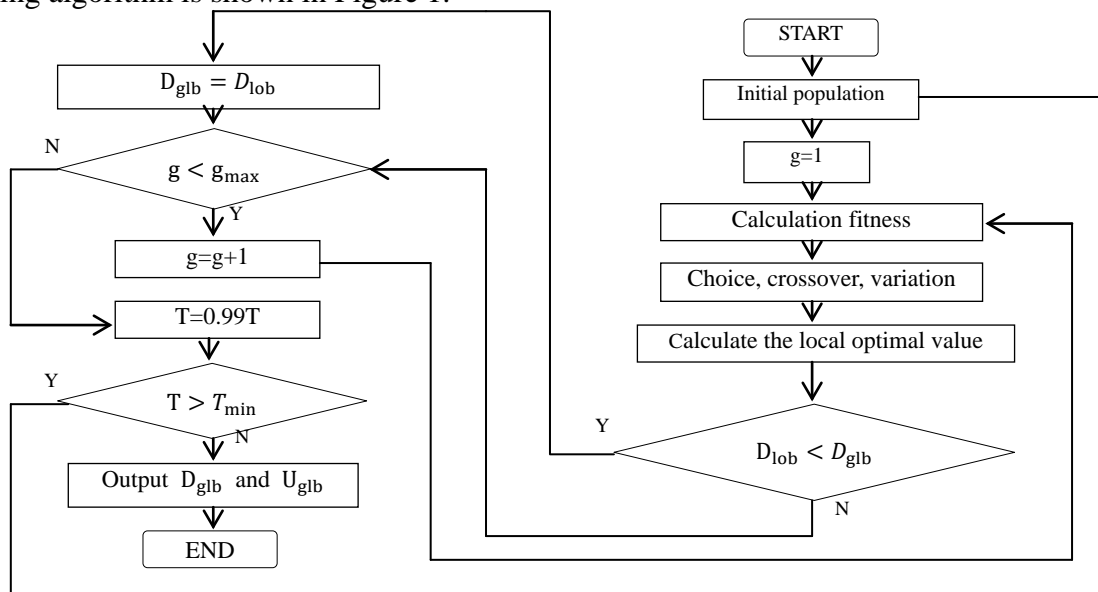


Figure 1 Flow chart based on GSA.

4. Experimental analysis

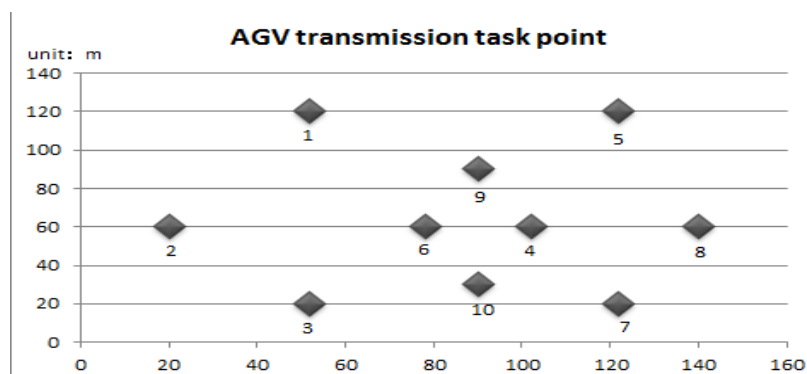


Figure 2 The AGV's route environment map model.

Taking Figure 2 as the graph model of the operating environment, suppose that there are ten task points that need to transport materials between V_1 and V_{10} , and then carry out path planning based on the genetic simulation of the combination of the non-combined genetic algorithm and the proposed paper. The annealing optimization algorithm performs optimal path comparison to verify the feasibility of the combined genetic simulated annealing algorithm. Solve the shortest path of the AGV after ten mission points, without considering the path constraint problem.

Based on the above theories and methods, the AGV optimal path problem of the intelligent factory is solved, and the simulation experiment is conducted through MATLAB R2014b. The maximum number of iterations is set as $g_{\max}=100$, population Size=20, crossover probability $P_C=0.5$, and mutation probability $P_M=0.15$. The algorithm part is as follows:

```

For k = 1:popsize
    If fitvalue(k) > newobjvalue(k)
        newpop(k,:) = pop(k,:);
    Else
        p = rand;
        If p <= exp((newobjvalue(k)-fitvalue(k))/t)
            End
        Pop = newpop;
    [bestindividual,bestfit]=best(pop,fitvalue);

```

Fitvalue is the fitness of the calculated population, newobjvalue is the target function value, and bestindividual is the most fitness of the population. Set $u = 1.25$ in the temperature reference value. The attenuation coefficient U_1 of jump probability P_1 is 5000., and the attenuation coefficient U_2 of jump probability P_2 is 5. Table 1 is obtained according to the comparison of the shortest distance between the genetic simulated annealing algorithm and the non-combined genetic algorithm:

Table 1 The results compared by uncombined algorithm between GSA.

	Non-combined algorithm	GSA	Percentage improvement
Shortest distance	460.81	444.08	3.6%
Shortest path order	10→7→8→4→9→5 →1→1→2→3	6→4→8→7→10→3 →2→1→5→9	

5. Conclusion

In this paper, an improved genetic simulated annealing algorithm is proposed to improve the fitness function in genetic algorithm, which effectively improves the ability of the algorithm to jump out of local optimum, and applies the improved idea to the path optimization problem of intelligent factory AGV. The shortest path relative to the non-combined genetic algorithm proves its feasibility. However, this paper does not consider the obstacle avoidance problem when multiple AGVs work. Later, we can combine multiple algorithms to explore better and more efficient solutions to find a more reasonable scheduling scheme.

References

- [1] Gao H C, Feng B Q, Zhu L. Reviews of meta-heuristicalgorithm for TSP[J]. Control and Decision, 2006, 21(3):241-247.
- [2] Ji J Z, Huang Z, Liu C N. A fast ant colonyoptimization algorithm for traveling salesman problems [J]. J of Computer Research and Development, 2009, 46(6):968-978.
- [3] Liu Z H, Zhang Y J, Zhang J, et al. Using combinationof ant algorithm and immune algorithm to solve TSP[J].Control and Decision, 2010, 25(5): 695-700.
- [4] Xiang Z Y, Liu Z C, Shen P A. An improved adaptivegenetic algorithm based on evolutionary stages [J]. J ofWuhan University: Engineering, 2008, 41(1): 133-136.
- [5] Peng D P, Lin Z Y, Wang J Q. An improved genetical gorithm for TSP problem [J]. Computer Engineeringand Applications, 2006, 42(13): 91-93.)
- [6] Lin W, Delgado-frias J G, Gause D C, et al. Hybrid newton-raphson genetic algorithm for the

traveling salesman problem[J]. *Cybernetics and Systems*, 1995, 26(4): 387-412.

[7] Maulik U, Bandyopadhyay S. Genetic algorithm-based clustering technique [J]. *Pattern Recognition*, 2000, 33(9):1455-1465.

[8] Lin W, Delgado-frias J G, Gause D C, et al. Hybridnewton-raphson genetic algorithm for the travelingsalesman problem[J]. *Cybernetics and Systems*, 1995, 26(4): 387-412.