ITM Web of Conferences **45**, 01017 (2022) https://doi.org/10.1051/itmconf/20224501017
CSCNS2021

Model and solution for the traveling salesman problem with multiple time windows

Xiaoling Zhang*, and *Yueli* Ni

Nanjing Institute of Tourism & Hospitality, Nanjing, Jiangsu, China.

Abstract. This paper applies the multi-time window traveling salesman problem to not only optimize the logistics cost, but also effectively endow users with multiple discrete idle optional time periods to meet the time requirements of just-in-time production. In the process of problem solving, firstly, the dynamic penalty factor is introduced into the objective function and the penalty function is added to relax the constraints of multi-time window in order to construct the relaxation model. Secondly, while in solving the model, the compressed annealing algorithm, which has the property of probability convergence, is proposed on the basis of the simulated annealing algorithm with only temperature parameter. The dynamic penalty factor is added as a pressure parameter to control the probability of the transition to an infeasible route regarding the time windows. Finally, comparison through data experiments between multi-time windows and single-time windows verifies the practicability of the former and comparison between the solution algorithm and simulated annealing algorithm verifies the stability of compressed annealing algorithm. The result shows that the compressed annealing algorithm is a comparatively better method to solve multi-time window traveling salesman problem.

1 Overview

With the development of lean production, the demand for punctuality is increasing and the timely and efficient services provided by the logistics industry are gaining more and more attention, so the Traveling Salesman Problem with time windows has been widely and intensively studied, among which the Traveling Salesman Problem with time window (TSPTW) is one of the important research directions. The practical applications of TSPTW in industry and service industry are also very wide: parcel delivery, bank courier, bus logistics and logistics handling systems with automatic guided vehicles. bank couriers, bus logistics and logistics handling systems with automated guided vehicles. In addition, the Traveling Salesman Problem with time windows is equivalent to the time sensitive production scheduling problem prevalent in the manufacturing industry.

The Traveling Salesman Problem with multiple time windows studied is an extension of the single time-window Traveling Salesman problem. Based on the time-constrained problem, the simulated annealing method is extended to the Traveling Salesman Problem with multiple time windows. The Traveling Salesman Problem with multiple Time windows is to find a

* Corresponding author: xiaoling-r@163.com

minimum-cost travel path in which a vehicle departs from a specific warehouse and returns to this warehouse, visiting one user at a time. Each user has multiple non-intersecting time windows, and the delivery vehicle must serve each user within or near a time window corresponding to that time window, and each user must be served. Multiple time windows can precisely meet the time requirements of users, and also enable users to effectively use multiple discontinuous free optional time slots, which can improve customer satisfaction and effectively reduce costs for enterprises, and have high practical value. Wolfler Calvo (2000) [1] proposed a heuristic algorithm which first used task relaxation to construct the initialization path and then improved the path by local search method. This heuristic is more effective for over 200 clients and instances with wide time windows. Ma, Zuo and Yand [2] studied the vehicle path problem with multiple time windows and established a mathematical model; Peng and Zhou [3] proposed to solve the vehicle path problem with multiple time windows. proposed a hybrid ant colony algorithm for solving the vehicle path problem with multiple time windows, and Huang and Li [4] designed a genetic algorithm for solving the vehicle path problem with multiple time windows. Huawei Ma [5] designed an improved ant colony algorithm for solving multi-time-window vehicle path problems that allows split distribution. Huang Qiu-ai et al [6] improved the traditional genetic algorithm by introducing the optimal individual retention mechanism and designed a mathematical model to solve it, which verified the effectiveness of the algorithm and its model. Zhu Ling-ling [7] proposed a collaborative contraindicated optimization algorithm, which solves by scanning algorithm to find the initial solution, and then design adaptive modification of contraindicated length and multiple sub contraindicated algorithms for collaborative optimization. Li Zhen-ping et al [8] used the intelligent water drop algorithm to explore the multi-time window vehicle path problem effectively.

In this paper, The Traveling Salesman Problem with multiple time windows is studied by introducing a dynamic penalty multiplier in the objective function, which relaxes the time constraint and establishes a relaxed mathematical model for The Traveling Salesman Problem with multiple time windows, which is solved by proposing a compressed annealing algorithm, which is a generalization of the simulated annealing algorithm to obtain an effective heuristic algorithm. Also the algorithm can maintain the property of probabilistic convergence to the global minimum set of solutions as already proved by Ohlmann.etal [9].

2. Problem analysis

The Traveling Salesman Problem with multiple time windows can be described as follows: a vehicle, starting from a distribution center, delivers to each user on the path and must serve each user. The users have multiple non-intersecting time windows, and the delivery vehicle must serve each user within (or near) one of the time windows corresponding to that user, and return to the distribution center after completing the delivery task. For the convenience of the study, it is assumed that the vehicle capacity is not limited and the vehicle service time for each user is negligible, i.e., it is set to 0. The vehicle arrives early and needs to wait for the time window to open before serving. Our goal is both to reduce the service outside the time window and to reduce the total delivery time.

Let $G=(N,A)$ be a finite graph, where $N=\{\mathbf{0},\mathbf{1},\ldots,n\}$ is the finite set of nodes or customers, A is the set of arcs connecting customers. A tour is defined by the order in which the n customers are visited and denoted by $\Im=\{p_{\mathbf{0}},p_{\mathbf{1}},\ldots,p_n,p_{n+\mathbf{1}}\}$, Let S be all path sets, where p_i denotes the index of the customer in the i th position of the tour. Let customer 0 denote the depot and assume that every tour begins and ends at the depot, i.e, $p_{\mathbf{0}}=p_{n+\mathbf{1}}=\mathbf{0}$.

Let $\Im = \{p_0, p_1, \ldots, p_n, p_{n+1}\} \in S$ be a tour, $c(p_i)$ denotes the travel time from p_i to p_{i+1}. Thus the total time to complete the path is $f(\Im) = \sum_{i=0}^{n} c(p_i)$ (1). a_{p_i} denotes the moment of arrival of the vehicle at point p_i, $\left[e_{mp_i}, l_{mp_i}\right]$ $m = 1, 2, \ldots, k_{p_i}$ denotes the k_{p_i} delivery time windows for the user at point p_i

The Traveling Salesman Problem with multiple time windows is composed of two parts: the Traveling Salesman Problem and the multi-time-window. The Traveling Salesman Problem is NP-hard, and the multiple time windows increase the complexity of the problem again. In this paper, we introduce the idea of penalty function to add the constraints of time windows to the objective function, which also facilitates the heuristic search for the solution.

In order to define the penalty function $P(\Im)$, it is necessary to consider four scenarios for the moment a_{p_i} when the vehicle reaches the user at point p_i on the path.

(1) If $a_{p_i} \le e_{1p_i}$, i.e., the vehicle arrives before the first time window opens for the user at point p_i and needs to wait, then $P(p_i) = e_{1p_i} - a_{p_i}$, so the moment of reaching the user at point p_{i+1} is $a_{p_{(i+1)}} = e_{1p_i} + c(p_i)$

(2) If $a_{p_i} \ge l_{k_{p_i} p_i}$, i.e., The vehicle arrives after the last time window closes for the user at point p_i, then $P(p_i) = a_{p_i} - l_{k_{p_i} p_i}$, so the moment of reaching the user at point p_{i+1} is $a_{p_{(i+1)}} = a_{p_i} + c(p_i)$.

(3) If $e_{jp_i} \le a_{p_i} \le l_{jp_i}$, $(j = 1, 2, \cdots k_{p_i})$, i.e., the vehicle arrives at the user at point p_i at the moment exactly in the j th time window, then $P(p_i) = 0$, so the moment of reaching the user at point p_{i+1} is $a_{p_{(i+1)}} = a_{p_i} + c(p_i)$.

(4) If $l_{jp_i} \le a_{p_i} \le e_{(j+1)p_i}$ $(j = 1, 2, \cdots k_{p_i} - 1)$, i.e., The vehicle arrives between the j th time window and the $j+1$ th time at point p_i.

If $a_{p_i} \ge \frac{e_{(j+1)p_i} + l_{jp_i}}{2}$, i.e., the moment Vehicle arrives at point p_i is closer to the $j+1$ th time window, can be considered early to need to wait for the $j+1$ th time window to open, then $P(p_i) = e_{(j+1)p_i} - a_{p_i}$, so the moment of reaching the user at point p_{i+1} is $a_{p_{(i+1)}} = e_{(j+1)p_i} + c(p_i)$.

If $a_{p_i} \le \frac{e_{(j+1)p_i} + l_{jp_i}}{2}$, i.e., the moment the vehicle arrives at the user at point p_i is closer to the j th time window and can be considered late to serve immediately, then $P(p_i) = a_{p_i} - l_{jp_i}$, so the moment of arriving at the user at point p_{i+1} is $a_{p_{(i+1)}} = a_{p_i} + c(p_i)$.

In summary, the time penalty function for the vehicle at point p_i is obtained as

$$P(p_i)=\begin{cases} e_{1p_i}-a_{p_i} & 当 a_{p_i}\le e_{1p_i} \\ a_{p_i}-l_{k_{p_i}p_i} & 当 a_{p_i}\ge l_{k_{p_i}p_i} \\ 0 & 当 e_{jp_i}\le a_{p_i}\le l_{jp_i},(j=1,2,\cdots k_{p_i}) \\ e_{(j+1)p_i}-a_{p_i} & 当 \frac{e_{(j+1)p_i}+l_{jp_i}}{2}\le a_{p_i}\le e_{(j+1)p_i},(j=1,2,\cdots k_{p_i}-1) \\ a_{p_i}-l_{jp_i} & 当 l_{jp_i}\le a_{p_i}\le \frac{e_{(j+1)p_i}+l_{jp_i}}{2},(j=1,2,\cdots k_{p_i}-1) \end{cases} \tag{2}$$

Thus the penalty function of the whole path is

$$P(\Im)=\sum_{i=1}^{n}P(p_i) \tag{3}$$

Also based on the above analysis the moment of arrival of the user at point p_{i+1} is obtained as follows:

$$a_{p_{i+1}}=\begin{cases} e_{1p_i}+c(p_i) & 当 a_{p_i}\le e_{1p_i} \\ e_{(j+1)p_i}+c(p_i) & 当 \frac{e_{(j+1)p_i}+l_{jp_i}}{2}\le a_{p_i}\le e_{(j+1)p_i},(j=1,2,\cdots k_{p_i}-1) \\ a_{p_i}+c(p_i) & other \end{cases} \tag{4}$$

3. Model formulation

The general traveling salesman r problem is modeled as follows. min $f(\Im)$ s.t. $\Im\in S$ where S is the set of feasible paths, $f(\Im)$ is determined by equation (1).

In this paper, the penalty function is added to the objective function, which can relax the restriction of the time window. A sufficiently large non-negative dynamic penalty factor b is introduced according to the literature [10] to make the relaxation model and the original planning model duality. Now, we build the following relaxation model for the multi-time window problem.

$$\min v(\Im,\lambda)=f(\Im)+\lambda P(\Im)$$

$$\text{s.t. } a_{p_i}+\sum_{m=1}^{k_{p_i}}\max\{(e_{mp_i}-a_{p_i})\operatorname{sgn}(a_{p_i}-\frac{1}{2}(e_{mp_i}+l_{(m-1)p_i}),0\}+c(b_i)=a_{p_{i+1}} \tag{5}$$

$l_{0p_i}=0, i=1,2,\dots,n$, $e_{mp_0}=0, m=1,2,\dots,k_{p_0}$, $p_i\in\{0,1,2,\cdots,n-1\}, i=0,\cdots,n-1$, $p_i\ne p_j, i,j=1,2,\cdots,n; i\ne j$, $p_0=0$, $p_{n+1}=0$

where $f(\Im),P(\Im)$, $\Im=\{p_0,p_1,\dots,p_n,p_{n+1}\}\in S$, are established by equations (1), (2) and (3), and the first constraint is a deformation of equation (4), which indicates the relationship between the moment of arrival of the vehicle at point p_{i+1} and the moment of arrival at point p_i.

4. Model solution

The compressed annealing algorithm is an improvement on the simulated annealing algorithm. The compressed annealing algorithm adds a dynamic penalty factor λ as a pressure to the simulated annealing algorithm, giving the algorithm two parameters. The temperature parameter controls the probability that the route cost becomes higher and the pressure parameter controls the probability that the arrival moment is not in the time window.

In a compressed annealing algorithm, the iterative correction of the pressure λ will involve the pressure valve λ^*. The approximate value of the pressure valve $\hat{\lambda}$ is generally obtained by multiple iterations. In this paper, we follow the approximation function of the pressure valve in the literature [2]. $\hat{\lambda}=\max_{\Im\in R}\{\frac{f(\Im)}{p(\Im)}\frac{\kappa}{1-\kappa}\}$, where the value of κ represents the percentage of penalty terms with respect to the objective function and $0\le\kappa\le 1$, R is a feasible path set.

During the search process of the compression annealing algorithm, the pressure is gradually increased, while the temperature is gradually decreased. Therefore, the initial value of the pressure can be determined as $\lambda_0=0$. The calculation of the initial value of temperature τ_0 and the approximation of the pressure valve $\hat{\lambda}$ requires the following steps.

(1) Generate random r pairs of adjacent feasible paths in S to form the set a. Compute the average absolute difference $|\Delta v|$ of the objective function $f(\Im)$ in the set R.

(2) The initial value of the temperature is $\tau_0=\frac{|\Delta v|}{\ln(\frac{1}{\chi_0})}$ (6), where $0.80\le\chi_0\le 0.99$. The approximate value of the pressure valve is $\hat{\lambda}=\max_{\Im\in R}\{\frac{f(\Im)}{p(\Im)}\frac{\kappa}{1-\kappa}\}$, the range of κ values in the numerical experiments of this paper is $[0.75, 0.99]$.

Let the length of Markov chain in the algorithm be l, the cooling schedule for each iteration be $\{\tau_0,\tau_1,\cdots\}$, and the pressurization schedule be $\{\lambda_0,\lambda_1,\cdots\}$. τ_k and λ_k are the temperature and pressure values when iterating from $kl+1$ to $(k+1)l$, respectively.

Referring to the cooling and Ohlmann [12] pressurization criteria in Dowsland [11], the chosen correction equation for cooling and pressurization is $\tau_{k+1}=\beta\tau_k$ (7)and $\lambda_{k+1}=\hat{\lambda}(1-\frac{(\hat{\lambda}-\lambda_0)}{\hat{\lambda}}e^{-\gamma k})$ (8)

The range of two of these parameters is $0.80\le\beta\le 0.99, 0.01\le\gamma\le 0.1, \lambda_0=0, \hat{\lambda}\ge 0$. Now, we give the two-stage heuristic algorithm for solving the relaxation model (5) designed based on the compressive annealing algorithm.

Algorithm 4.1

Step 1 Gives the initial conditions.

(1) Initialize the optimal path $\Im_{best}$ such that $f(\Im_{best})=\infty$ and $P(\Im_{best})=0$; initialize the parameter $m=0$, choose the Markov chain length as l, and do not accept the upper bound $I_{\max}$ on the number of feasible solutions .

(2) Randomly generate feasible solutions $\Im_0$, $k=\mathbf{0}$, τ_0 is established by equation (6).

(3) $\mathbf{0.80} \le \beta \le \mathbf{0.99}, \mathbf{0.01} \le \gamma \le \mathbf{0.1}, \ \lambda_0 = \mathbf{0}, \hat{\lambda} \ge \mathbf{0}$

Step 2 Sub-iterations in step k .

The temperature τ_k and pressure λ_k are unchanged, $st=\mathbf{0}$.

A local search in the neighborhood of $\Im_k$ search to generates a new feasible solution y .

(2) If $v(y,\lambda_k) < v(\Im_k,\lambda_k)$,then $\Im_k = y$, $st = st+\mathbf{1}$.

If $v(y,\lambda_k) > v(\Im_k,\lambda_k)$ and $\exp(\frac{-(v(y,\lambda_k)-v(\Im_k,\lambda_k))^+}{\tau_k}) > rand(\mathbf{0},\mathbf{1}]$,then $\Im_k = y$.

(3) If $P(\Im_k) \le P(\Im_{best})$ and $f(\Im_k) < f(\Im_{best})$,then $\Im_{best} = \Im_k$; otherwise $m = m+\mathbf{1}$.

(4) If $st \le l$,return Step2(1); otherwise go toStep3.

Step 3 Determine the end condition.

Correct τ_k and λ_k by (7) and (8); $k=k+\mathbf{1}$;If $m \ge I_{\max}$, go to Step 4, otherwise go to Step 2.

Step 4 The optimal solution is output and the computation is finished. Probabilistic convergence of Algorithm 4.1 is obtained according to Hajek B. [13] and Ohlmann etal [9].

Theorem 1 If Algorithm 4.1 satisfies.

(1) Markov chain is integrable

(2) $\lim_{k\to\infty} \tau_k = \mathbf{0}$, $\lim_{k\to\infty} \lambda_k = \bar{\lambda} > \lambda^*$, φ^* is the global optimal solution set .

while $\sum_{k=\mathbf{0}}^{\infty} \sum_{x,y\in S_k} \exp(-[v(y,\lambda_k)-v(x,\lambda_k)]^+ / \tau_k) = +\infty$ holds, where S_k is the set of all accepted update paths in Step 2 of Algorithm 4.1, then $\lim_{k\to\infty} P\{y_k \in \varphi^*\} = \mathbf{1}$ (9)

5. Analysis of numerical results

Because the data of real logistics enterprises are not available and there is no standard test data of multi-time-window path problem in the international arena, the test data of this paper is composed by constructing 100 random nodes with the coordinates of nodes in [10, 50] according to random distribution, the coordinates of the starting point are the middle position (30,30), the starting point has no time window, and the remaining nodes are randomly generated in [0,12] respectively 1, 2, 3 mutually non-coincident time windows, the total time sum of each time window does not exceed 6. In this paper, the relevant parameters in the multi-time-window travel quotient problem solved by the compressed annealing algorithm take the values $\beta = \mathbf{0.95}, \gamma = \mathbf{0.06}$, $\chi_0 = \mathbf{0.94}$, $\kappa = \mathbf{0.9999}$, $l = \mathbf{3000}$, $I_{\max} = \mathbf{75}$.

The algorithm was implemented in Win7 (Intel core i3-2370M CPU2.4GHZ, 6Gmemory) platform using matalab2016a language. The data experiments were tested for 20, 50, 80, and 100 nodes, respectively, and the minimum and average cost results of the paths were obtained for each set of data tested 100 times as shown in Table 1. the results of solving the test data for 100 nodes using the simulated annealing algorithm and Algorithm 4.1, respectively, are shown in Table 2.

From the analysis of Table 1, it is obtained that the minimum cost gradually decreases with the increase of the number of time windows, which is consistent with the reality. Meanwhile, the average value of each group of data is closer to the minimum value, and the stability of Algorithm 4.1 is better. As can be seen from Table 2, the difference between the minimum values of the two algorithms is not large, but the mean value of the simulated annealing algorithm deviates more from the minimum value, which is less stable. In terms of

CPU running time, Algorithm 4.1 is significantly faster than the simulated annealing algorithm. In a comprehensive comparison, Algorithm 4.1 can effectively solve the Traveling Salesman Problem with multiple time windows.

Table 1. Algorithm 4.1 Experimental results.

	20groups		50groups		80 groups		100groups	
	Min.	Average	Min.	Average	Min.	Average	Min.	Average
1 time window	193.92	214.63	373.64	430.94	673.60	719.47	874.88	943.26
2 time windows	190.65	207.44	371.45	425.51	658.72	706.90	853.61	922.02
3 time windows	179.58	183.40	315.75	369.75	636.87	686.15	807.95	895.98

Table 2. Comparison of the results of simulated annealing algorithm and Algorithm 4.1

Min.	Min.				Algorithm 4.1			
	Min.	Average	CPU(s) Min.	CPU(s) Average	Min.	Average	CPU(s) Min.	CPU(s) Average
1time window	877.45	1030.92	9.98	13.13	874.88	943.26	2.86	2.92
2time windows	827.53	948.31	10.82	16.44	853.61	922.02	3.15	3.19
3time windows	810.46	902.48	10.53	17.50	807.95	895.98	3.72	3.84

6. Conclusion

We studied the Traveling Salesman Problem with multiple time windows, built a relaxation model for the problem, and then designed a two-stage heuristic algorithm based on the compressed annealing algorithm to solve the problem, and the experimental results showed that the algorithm can effectively solve the traveler problem with multiple time windows.

In this paper, we only consider the impact of multi-time window constraints on the model, but the impact of real-life constraints such as the number of vehicles, the maximum distance limit, the maximum service time limit, and the vehicle capacity on the model is not further investigated, while the simulated data used in the data experiments. In further research work, we can consider introducing more constraints and obtaining real data of enterprises, so that the theory of the Traveling Salesman Problem with Multiple Time Windows can serve logistics enterprises better.

Jiangsu Province Key Research Base of Philosophy and Social Science in Universities "Research Center for Applied Tourism Talents in the New Era”

References

1. Wolfler Calvo, R. A new heuristic for the traveling salesman problem with time windows. Transportation Sci. (2000).
2. Ma H, Zuo C, Y and S. Modeling and solving for vehicle routing problem with multiple time windows. Journal of System Engineering, (2009).
3. Peng B, Zhou Y. Hybrid ant colony algorithm for vehicle routing problem with multiple time window. Computer Engineering and Applications,(2010).

4. Huang Q, Li Z .Mathematical model and algorithm for vehicle routing problem with multiple time windows. Logistics Technology,(2012).
5. Ma, H. W., Ye, H. R., Xia, W.. Improved ant colony algorithm for solving multi-time window vehicle scheduling problem allowing split distribution. China Management Science. (2012).
6. Huang Qiu-Ai, Li Zhen-Ping. Mathematical model and algorithm for multi-time window vehicle path problem. Logistics Technology. (2012).
7. Zhu L-L, Yang A-Q, Wu Kuan-Ren. Implementation of multi-temporal window VRP algorithm based on cooperative adaptive forbidden. Computer Application Research. (2012).
8. Li ZP, Zhao F, Liu HW. Intelligent water drop algorithm for multi-time window vehicle path problem. Operations Research and Management. (2015).
9. Ohlmann J, Henderson B S. Convergence in probability of compressed annealing. Math. of Oper. Res. (2004).
10. Hadj-Alouane A, Bean J. A genetic algorithm for the multiple-choice integer program. Oper. Res. (1997).
11. Dowsland K A. Simulated Annealing. John Wiley and Sons, New York. (1993).
12. Ohlmann J W, Thomas B W. A compressed annealing heuristic for the traveling salesman problem with time windows. Informs Journal on Computing.(2017)
13. Hajek B. Cooling schedules for optimal annealing. Mathematics of Operations Research,(1988)