Variable Neighborhood Search with Hill-Climbing Algorithm for Large Scale Optimization

Xueshi Dong[*1,2], Fanfan Shen[3], Lingling Zhang[4], Zhenghao Xu[1,5], Yongchang Shan[1], Qing Lin[1,6]
[1]College of Computer Science and Technology, Qingdao University, Qingdao, 266071, China
[2]Shandong Provincial Key Laboratory of Fundamental Software, Jinan 250101, China
[3]School of Computer Science, Nanjing Audit University, Nanjing 211815, China
[4]Business School, Shandong University, Weihai, 264200, China
[5]Anhui Provincial Key Laboratory of Multimodal Cognitive Computation, Anhui University, Hefei, 230093, China.
[6]State Key Laboratory of Public Big Data, Guizhou University, Guiyang, 550025, China
[*]Corresponding author E-mail: dxs_cs@163.com

Abstract: Multiple maximum scatter traveling salesperson problem (MMSTSP) can be used to model the applications such as maximizing the benefits of multiple workflows with multiple machines, the scheduling optimization of logistics and manufacturing with multiple vehicles (machines). In computer science and operation research, MMSTSP are utilized to simulate some applications, where the scale easily tends to large scale. However, the traditional algorithms for large scale optimization are still limited on solution quality and convergence speed, these algorithms are based on heuristic rules and usually lack mathematical theoretical support. Aiming at this problem, this paper gives a novel hybrid variable neighborhood search based on hill-climbing algorithm (VNSHA) as the local optimization with Wiener process for solving large scale MMSTSP. The solutions of MMSTSP are optimized by the initialization operator, where the cities are inserted into the tours with the distance as max as possible, then the variable neighborhood search (VNS) and an extended hill-climbing algorithm (Interior and external neighborhood search optimization) with Wiener process are used for further optimization. During this process, the traveling routes can be optimized by changing the cities visiting sequence based on Wiener process. Experiments can show that the proposed algorithm VNSHA demonstrates better solution quality than the state-of-the-art algorithms for large scale optimization of MMSTSP.
Keywords: Variable neighborhood search, large scale optimization, neighborhood optimization, multiple maximum scatter traveling salesperson problem, Wiener process

1 Introduction

Traveling salesman problem (TSP) is non-deterministic polynomial (NP) hard problem in combinatorial optimization, which is important in operational research and theoretical computer science. TSP is a classical combinatorial optimization problem (COP). Because of its wide application in transportation, circuit board design and logistics distribution, scholars at home and abroad have done many studies on it. The studies not only focus on solving methods, but also on its variants. In the recent past years, many versions of TSP, such as multiple traveling salesman problem (MTSP), colored traveling salesman problem (CTSP) [1] and multiple maximum scatter traveling salesperson problem (MMSTSP), are proposed and applied in many fields. They provides an ideal platform for studying COP. Many COP problems, such as knapsack problem and job shop scheduling problem, are NP complete problems with the same difficulty as TSP versions. If one of them can be solved by polynomial deterministic algorithms, all other NP complete problems can also be solved by polynomial algorithms. Many methods were originally developed from TSP variants and later extended to other NP complete optimization problems.

Maximum scatter traveling salesperson problem (MSTSP) [2] belongs to a version of basic traveling salesman problem TSP, its goal or task is to maximize the min edge of the traveling route for a salesman, it has been used in assembly line sorting and workflow scheduling. Since MSTSP contains a salesman and its task, it cannot be applied to the applications with multiple tasks, thus this paper discusses another combinational optimization model MMSTSP, which can model these kinds of problems. Dong et al. [3] discussed the MMSTSP and its related works, and applied genetic algorithm (GA), greedy GA (GAG), climb-hilling GA (HCGA) and simulated annealing GA (SAGA) for MMSTSP, and lots of experiments proved the effectiveness. Because MMSTSP belongs to NP-hard problem, the mentioned traditional intelligent algorithms can be used for solving it. However, these algorithms are easy to fall into local optimum, the convergence speed is needed to be improved, and the scale of problem model is also limited. Therefore, it is necessary to design better algorithms for solving large scale optimization of MMSTSP problem.

Table 1 The statistical information of researches at home and abroad

Contents	Literature	Date of Publication	Numbers
Relevant works of TSP versions	[1-6]	2015,2010,2017,2013,2018,2019	6
Large scale optimization	[7-14]	2015,2017,2019,2022,2024	8
Variable neighborhood search	[15-19]	2018,2021	5
Advanced methods for optimization problems	[20-24]	2020,2021	5
Other related or extended works of this research	[25-42]	2015,2018,2020,2021,2022,2023,2024,2025	18

Table 1 is the statistical information of the subject at hand, the supporting references for the statistical information are provided.

In recent years, relevant works of TSP versions are as follows: Li et al. [1] used the improved GA algorithms, including SAGA, GAG and HCGA for solving CTSP, it is an extended model of the basic TSP, however, the solving algorithms are limited on performance [1]; the Ref.[2] discussed some versions of TSP such as bottleneck TSP (BTSP) and multiple traveling salesman problem MTSP, and gave traditional methods for solving these problems; after that, Ahmed applied a hybrid GA algorithm based on optimization strategy to solve the BTSP problem, but the scale of BTSP mainly focus on small scale optimization [4]; Dong et al. [5] presented improved ant colony optimization (ACO) to solve CTSP using multiple tasks learning, and extend the scale of CTSP to large scale where the city number is more than 1000; Dong and Cai [6] provided a new GA algorithm using ITÖ process (Wiener process) for large scale optimization of colored balanced TSP (CBTSP). Although many intelligent algorithms are proposed to solve the kinds of TSP versions, there are still improvement space for the performance of the algorithms.

Furthermore, many works have done on large scale optimization: the improved optimization algorithms, such as particle swarm optimization [7], differential evolution [8], competitive swarm optimizer [9], were utilized for large scale optimization problems, and the researchers have also proved the effectiveness of these algorithms, but there are still limitations on performance. In this study of Ref.[7], it proposed a greedy discrete particle swarm optimization framework for large-scale social network clustering, where particle states are redefined in the discrete scenario, and the status update rules is based on the network topology; in the Ref.[8], two different weaknesses were re-examined and a plan was developed, which can increase the diversity of test vectors and improve the exploration ability of differential evolution for large-scale optimization; Dong et al. [10] gave a hybrid ITÖ algorithm to solve large scale CTSP, where the max city number is more than 10000; for large scale multi-objective optimization, Ref.[11] proposed a hierarchical multi-strategy learning group algorithm (LSLSO). The optimization process of LSLSO is divided into hierarchical multi-strategy search and detailed search, the particle is divided into four levels according to its fitness value. Furthermore, the optimized machine learning algorithms, including differential grouping algorithm [12], segment-based predominant learning [13], semi-definite relaxation [14], were also applied to solve large scale optimization problems.

Moreover, the recent works for variable neighborhood search are discussed: Qiu et al. [15] gave a variable neighborhood search (VNS) for optimizing production routing problems, where the routing decisions are done by local search using variable neighborhood strategies; Hore et al. [16] applied the improved VNS for studying the basic TSP, where a stochastic approach is used to find the optimal solution; an improved ITÖ optimization algorithm based on the local search strategy was proposed for studying the large scale optimization of multiple bottleneck TSP, where a deleting and reinserting operator for city sequences is used in this process [17]; Dong et al. [18] applied VNS to improve hybrid genetic algorithm for multi-scale optimization of multiple bottleneck TSP problem; Xu et al. [19] gave an improved VNS for solving large scale optimization of the general CTSP, where a delaunay-triangulation-based optimization operator is integrated into VNS. From the literatures, it shows that VNS and its improved versions can display good performance for large scale optimization of TSP variants.

In addition to the above discussions, there are other broader and more general discussions, where highlights the importance of advanced solution methods for various optimization problems (e.g., online learning, scheduling, network design, and others): Pasha et al. [20] gave an optimization algorithm by integrating environmental considerations and heterogeneous ship fleet for liner shipping planning problem; Liu et al. [21] proposed an evolutionary many-objective optimization method by density estimation and angle-based selection for a variety of optimization problems, lots of experiments showed the effectiveness of the given algorithm; Dulebenets [22]

designed a memetic algorithm based on an adaptive polyploid strategy for trucks scheduling problem; Fathollahi-Fard et al. [23] provided two hybrid nature-inspired algorithms based on the optimization algorithms such as whale optimization algorithm and genetic algorithm for solving the design problem of supply chain network; Rodrigues-Jr et al. [24] discussed a trajectory prediction algorithm, where a bidirectional minimum-gated recursive network was used for effective patient trajectory prediction.

Additionally, other related or extended works are as follows: Ref.[25] proposed a memetic algorithm (MA) to handle the interval multi-objective optimization problems, where the increment of the super volume is used to formulate an activation policy for each local search process; in Ref.[26], the authors examined self-attention along all possible dimensions, explored all possible aggregations of self-attention, and then applied neural architecture search (NAS) to achieve optimal aggregation; based on a new greedy strategy, a greedy repair and optimization algorithm is proposed to eliminate infeasible solutions, and a general discrete evolutionary algorithms framework is proposed for hardware/software problems [27]; in this paper, the researchers proposed a new method for knowledge distillation, aiming to take full advantage of the similarity between multiple categories, and they also proposed a hybrid technique to better capture the similarity correlation between different instances [28]; an adaptive large neighborhood search algorithm was proposed to solve the dynamic vehicle routing problem, where temporary destroy/repair trackers and periodic disturbance procedures are used in this work [29]; in this paper, a hybrid optimization method based on biogeography and variable neighborhood search was implemented, where improved nearest neighbor mechanisms were used to generate potential initial populations, and a hybrid migration operator was designed [30]; Ref.[31] introduced a ridge regression method for supervised classification, which can estimate a representation model, while taking into account the differences between classes, so that the classification information can be accurately derived; in this paper [32], the authors proposed a new model based on matrix trifactorization, which can reduce the complexity to linearity and make it fully scalable, and overcome the shortcomings of the most methods; Ref.[33] proposed an iterative approach based on VNS which used tree search in its local neighborhood exploration, it can perform several neighborhood explorations by controlling two parameters; the researchers [34] used advantage of recent advances in training deep neural networks to develop a new AI agent, called the deep Q network, which can use end-to-end reinforcement learning to learn successful strategies directly from high-dimensional sensory input; Ref.[35] presented a hybrid heuristic algorithm in which the initial solution was created by the optimal TSP solution, where an implementation of common variable neighborhood search was used to obtain delivery routes for trucks and drones; in Ref. [36], an augmented variable neighborhood search was proposed as the main framework for solving precedence-constrained CTSP, the variation of the exchange is combined with multiple insertions; in Ref. [37], a dynamic multi-objective evolutionary algorithm based on the multi-strategy and hybrid change response was given for solving dynamic multi-objective optimization problems; Zhou et al. [38] gave a Bi-trajectory hybrid search for bottleneck-minimized CTSP, where a crossover is used to generate offspring solutions, a multi-neighborhood simulated annealing is for local optimization; Zhang et al. [39] applied deep reinforcement learning to solve dynamic traveling salesman problems; Zhou et al. [40] used a multi-neighborhood simulated annealing-based iterated local search to solve colored TSP, where an edge assembly crossover is used for optimization; Dong et al. [41] proposed a hybrid ITÖ algorithm based on improved crossover and mutation operators for maximum scatter CTSP (MSCTSP); Dong et al. [42] gave a hybrid genetic algorithm with Wiener process, where the crossover and mutation with Wiener process are used for optimization.

From the mentioned studies, there are the following trends: 1) more and more combinational optimization models such as CTSP and MMSTSP are proposed to model the real-world problems; 2) the nature-inspired algorithms including variable neighborhood search and genetic algorithm are widely used for solving TSP and its versions; 3) large scale optimization is a hot research topic for the variants of TSP. According to these trends, the motives or objectives are as follows: 1) Although there are applications of MMSTSP, the applications are still limited, to extend more applications is the one of the motivations or objectives; 2) variable neighborhood search, genetic algorithm and their versions are applied to TSP variants, but the solution quality or solving speed is limited, it is needed to improve it; 3) the scale of models by MMSTSP is easy to large scale, therefor it is necessary to study the large scale optimization of MMSTSP problem.

Although the above developments have been achieved, the used solving algorithms for MMSTSP still have limitations on solving speed and solution. The gap or problem is that while

traditional algorithms solve MMSTSP, their solution quality and the scale of MMSTSP are limited, these algorithms are based on heuristic rules and lack mathematical theoretical support. Therefore studying better algorithms for large scale optimization is necessary and meaningful. For improving this problem, this paper presents a variable neighborhood search by hill-climbing algorithm (VNSHA) based on ITÖ process for large scale MMSTSP. For VNSHA, the solutions are optimized by variable neighborhood search and hill-climbing algorithm with ITÖ process. The experiments prove that VNSHA displays superiority over former approaches.

The innovations of the article include the six points: 1) Wiener process (ITÖ process) is introduced into variable neighborhood search (VNS) to provide the basic theory of mathematics and physics for the research in this field; 2) Gene coding is applied to multiple maximum scatter traveling salesperson problems (MMSTSP), and the gene computing method and optimization strategy based on VNS are designed; 3) The large-scale optimization for MMSTSP is studied for the first time; 4) A hybrid variable neighborhood search (VNSHA) is proposed for large scale MMSTSP; 5) An improved and extended hill-climbing algorithm (HA) with Wiener process is integrated into VNSHA algorithm for optimization; 6) Interior hill-climbing optimization and external hill-climbing optimization with Wiener process are firstly used in HA for optimizing.

The other sections are given: section II introduces MMSTSP and related works; section III is details of VNSHA for solving MMSTSP; section IV demonstrates the experiments and analysis; the fifth section concludes this paper and next works.

2 Relevant works

2.1 MMSTSP

A MMSTSP [3] contains n cities and m salespersons ($m<n$; $m, n \in Z$, $Z=\{1,2,3,...\}$). $G(V, E)$ stands for cities set, weight w_{ij} for edge $(i, j) \in E$, $i \neq j$, denotes the distance from cities i to j. V includes m city sets, the set V_i, $i \in Z_m$; $Z_m = \{1,2,3,\ldots m\}$is that it can be accessed by salesman i. d_i represents the starting and ending (depot). The objective function is to select m routes in G with the min edge as max as possible. MMSTSP is defined by the below formula.

$$\text{Max} \min\left\{C_{i,j}:(i,j)\square E, i\neq j\right\} \tag{1}$$

where the $C_{i,j}$ is the matrix for the route with the weight w, and (i, j) is the cost from city i to city j.

The x_{ijk} ($i, j \in V$; $i \neq j$; $k \in Z_m$) is if k_{th} salesperson is from city i to city j. The restrictive conditions [1,3] are:

$$\sum_{i=1}^{n-1} x_{0ik} = \sum_{i=1}^{n-1} x_{i0k} = 1, k \in Z_m \tag{2}$$

The second formula means that the depot is the ending and starting position;

$$\sum_{i}\sum_{j} x_{ijk} = \sum_{i}\sum_{j} x_{jik} = 0, i \in V_k, j \in V \setminus \{Depot \bigcup V_k\}, k \in Z \tag{3}$$

The third formula is that other traveler's exclusive cities can be visited by salesman k, and k_{th}'s exclusive cities are not allowed to be accessed by other traveling salespersons;

$$\sum_{j=0}^{n-1}\sum_{k=1}^{m} x_{ijk} = \sum_{j=0}^{n-1}\sum_{k=1}^{m} x_{jik} = 1, j \neq i, i \in V \setminus \{0\} \tag{4}$$

The fourth formula means that any city can be accessed exactly once.

$$\sum_{i}\sum_{j} x_{ijl} = \sum_{i}\sum_{j} x_{jil} = 0, i \neq j, k \neq l, i \in V_k, j \in V, l \in Z_m \tag{5}$$

Formula (5) indicates that traveler l ($\neq k$) neither starts with nor returns to the k_{th} city data set.

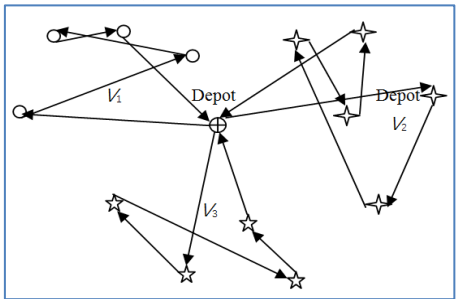

Fig.1. An example of MMSTSP

Figure 1 is an example of MMSTSP with 3 salespersons and 14 cities, V_1, V_2 and V_3 are three exclusive city sets, depot is ending and starting point, figure 1 is the simulating route by using MMSTSP instance data.

2.2 MMSTSP and MTSP

MMSTSP and MTSP (multiple traveling salesman problem): each city of them can be visited by salespersons only once, they can be transformed into other TSP variants by certain conditions; there are distinct objective functions for MMSTSP and MTSP.

In figure 2, the left small figure is an example of MTSP, its objective is finding a tour where the cost is as min as possible, simulation route is given in the left figure; the right figure is an example of MMSTSP, its objective is finding a tour with min edge as max as possible, and simulation path is given in the right small figure.

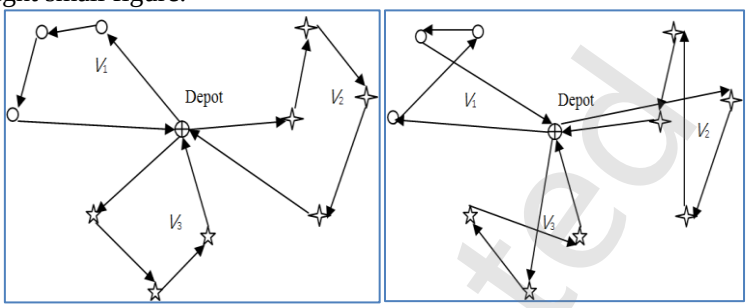

Fig.2. The examples of MTSP and MMSTSP

2.3 MMSTSP applications

MMSTSP can be utilized to model the applications such as the optimization of manufacturing with multiple machines, maximizing the benefits of multiple tasks or workflows with multiple machines. In the scheduling for tasks or workflows, it is often necessary to schedule based on the current maximum waiting time or maximum efficiency. A machine needs to handle multiple tasks or workflows, and in actual production processes, multiple machines often need to collaborate to complete production tasks or workflows. Therefore, this problem is a scheduling problem with multiple tasks and machines, and can be modeled by using MMSTSP.

According to the former literatures, MSTSP is maximizing min edge, and is utilized in the medical imaging and manufacturing. As an application of MSTSP [5], a falsely accused crime is to face penalty, he begins a journey for avoiding capture. With his wanted information such as name and pictures splashed across television, generally speaking, if he stays in a same place too long, he would easily rouse suspicion of the locals, and some ones may report him to the police. With the help of friends scattered across country who believe in his innocence, he will rotate from safe place to safe place in such a way that he is as far away from the previous safe place as possible. This instance is considered to be MSTSP. MMSTSP can be applied in the applications with multiple objectives such as the optimization of multiple crimes in this instance.

2.4 The theory of MMSTSP

Theorem 1. MMSTSP is NP-hard problem

Proof. MTSP can be converted to MMSTSP, if its objective function of each TSP is changed to maximize the difference between the maximum and minimum edge of multiple traveling routes. Since MTSP has been proved to be NP-hard problem, by changing the objective function or the conditions, its time complexity is not changed by this operation, thus MMSTSP is also NP-hard.

2.5 MMSTSP and MSCTSP

The similarities and differences between MMSTSP and MSCTSP [41] are as follows: they both have same objective function and multiple salesmen, are versions of TSP and MTSP; MMSTSP is a different problem with MSCTSP with distinct features such as the different access restrictions of salesmen, each salesman in MMSTSP can only visit own exclusive cities set, each salesman in MSCTSP not only access own exclusive cities sets, but also can visit shared city set.

3 VNSHA algorithm for large scale MMSTSP

3.1 Wiener process

The Wiener process [42] was first proposed by mathematician Norbert Wiener in the 1920s and is a mathematical model for describing stochastic analysis. Wiener process is as important in pure mathematics as in applied mathematics. In pure mathematics, the Wiener process leads to the study of continuous martingale theory, which is a basic tool to describe a series of important complex processes. It is indispensable in the research of stochastic analysis, diffusion process and

potential theory. In applied mathematics, Wiener process can describe the integral form of Gaussian white noise. In electronic engineering, Wiener process is an important part of establishing mathematical model of noise. Wiener process is closely related to Brownian motion in physics. Brownian motion refers to the endless random motion of pollen particles suspended in liquid. Wiener motion can also describe other random motions determined by Fokker Planck equation and Langevin equation. Wiener process forms the basis of rigorous path integral expression of quantum mechanics (according to Feynman Katz formula, the solution of Schrodinger equation can be expressed by Wiener process). In financial mathematics, Wiener process can be used to describe option pricing models such as Black Scholes (BS) model.

Definition: If a random process {X(t), t>=0}satisfies: (1)$X(t)$ is an independent incremental process; (2)Any s, t>0,$X(s+t)-X(s)\sim N(0,\sigma^{2t})$, that is a normal distribution, $X(s+t)-X(s)$ is expected to be 0, and the variance is σ^{2t}; (3)$X(t)$ is a continuous function with respect to t. Then {X (t), t>=0} is called Wiener process or Brownian motion.

Characteristics: Wiener process, also known as Brownian motion, has the following characteristics: The current value of the process is all the information needed to make its future prediction; Wiener process has independent increment. The probability distribution of the process change in any time interval is independent of the probability of its change in any other time interval; its variation in any finite time follows a normal distribution and increases linearly with the length of the time interval.

Given the Second-order moment process {$W(t)$, t>=0}, if it satisfies: it has independent increment; for any t>s>=0, the increment $W(t)-W(s)\sim N(0,\sigma^{2(t-s)}$, and s>0; W (0)=0. Then this process is called Wiener process.

The distribution of Wiener process increments is only related to the time difference, so it is a homogeneous independent increment process. It is also a normal process and its distribution is completely determined by its mean function and auto covariance function. Wiener process is not only a mathematical model of Brownian motion, but also the thermal noise of electronic components at constant temperature can be attributed to Wiener process. In the BS model of futures pricing model, futures prices and the underlying asset prices on which they depend are affected by the same uncertainty, and both follow the same Wiener process.

For any positive real number, the one-dimensional Wiener process is a random variable at any time, and its probability density function is:

$$f_{W_t}(x)=\frac{1}{\sqrt{2\pi t}}e^{-x^2/2t} \tag{6}$$

This is because according to the definition of Wiener process, when s=0, the distribution of w_t can be deduced: W_t = W_t – W_o~N(0, t). Its mathematical expectation is zero: $\mathrm{E}(W_t)=0$. Its variance is: t: $\mathrm{Var}(W_t)=E(W_t{}^2)-E^2(W_t)=E(W_t{}^2)-0=E(W_t{}^2)=t$, in the independent increment definition of Wiener process, let $t_2=t, s_2=t_1=s<t, s_1=0$, then $W_s=W_t-W_{S1}\sim N(0,s)$and $W_t-W_s=W_{t_2}-W_{s_2}\sim N(0,t-s)$ are independent random variables.

So at two different times 0<=s, t, the correlation coefficient with covariance and correlation function of W_t and W_s is:$\mathrm{cov}(W_s,W_t)=\min(s,t)$.

3.2 ITÖ process

ITÖ process can be seen as a generalized Wiener process, which directly means Brownian motion as random interference, thus it gives Brownian motion the most general meaning [42].

Stochastic differential equation (SDE) is:

$$dX_t=b(t,X_t)dt+\sigma(t,X_t)dB_t \tag{7}$$

Its integral form is:

$$X_t=X_0+\int_0^t b(t,X_t)dt+\int_0^t \sigma(t,X_t)dB_t \tag{8}$$

The solution of this equation X_t is called ITÖ process, or diffusion process. Under certain conditions, the solution of stochastic differential equation is unique.

$$L=\frac{1}{2}\sum_{i,j=1}^{N}a_{ij}(x)\partial_{ij}^{x}+\sum_{i=1}^{N}b_i(x)\partial_i^x \tag{9}$$

It can be obtained from ITÖ formula $f\in C_b^2(R^N)$;$(f(X_t)-\int_0^t Lf(X_u)du,F_t,P)$ is a martingale. It establishes the relationship between the diffusion process (X_t) and the second order differential operator L. The stochastic representation of parabolic equations can be given.

Weiner, the inventor of cybernetics, pointed out in 1923 that Brownian motion was a stochastic process in mathematics, and proposed to be described by "stochastic differential equation", so people also call Brownian motion as Weiner process. Mathematician ITÖ developed and established a stochastic differential equation with the interference term of Brownian motion,

$$dx(t) = \mu(t,x)dt + \sigma(t,x)dB \tag{10}$$

where μ (t, x) is the drift intensity, $\sigma(t, x)$ stands for the interference intensity, $\sigma(t, x)dz$ follows the normal distribution $N(0, \sigma^2(t, x))$. The activity intensity for variable neighborhood solution can be the drift intensity, and activity intensity for hill-climbing algorithm is considered to be the interference intensity. The process described by this equation is ITÖ process. ITÖ process can be seen as a generalized Wiener process, giving Brownian motion the most general meaning.

On this basis, economists used ITÖ process equation to describe the behavior process of stock prices. To more accurately describe the behavior process of stock prices, ITÖ process equation is modified to $dS(t)/S(t) = \mu dt + \sigma dB$, among them σ is the stock price volatility μ, the expected rate of return stock price is called the stock price equation, which is a stochastic differential equation. The stock price equation described by ITÖ process is a positive stochastic differential equation. Starting from the determined $S(0)=S_0$, the statistical behavior of the trajectory can be inferred according to the shape of the random variable B (t) of Brownian motion between 0 and t.

3.3 Solution coding and initialization

The solutions of MMSTSP are represented by the genes coding, the first sequence named city genes stands for the visiting cities sequence, the second one named salesperson genes is the salesman sequence. From figure 3, it shows that visiting cities of salesperson 2 are 4-11-7-8-5-6.

City Genes

4	11	1	7	2	9	8	3	5	10	6

Salesmen Genes

2	2	1	2	1	1	2	1	2	1	2

Fig.3. Solution of MMSTSP by genes coding

The initial solutions are important for performance of algorithms, the good solutions can make the algorithms obtain satisfactory results. In the initialization, the cities are added into the positions of the traveling with the route distance as max as possible.

The insertions for feasible solutions is based on a cost (distance) function. For a city k during this process, city i and city j are the two adjacent cities of this route, the distance for inserting city k between i and j can be computed by:

$$G(i, k, j) = d(i, k, j) = d_{ik} + d_{kj} - d_{ij} \tag{11}$$

where $d(i, k, j)$ is the additional (extra) distance by this inserting of city k.

It checks each possible insertion location of each candidate city. The city with the biggest insertion distance is in the best position. It is defined by the following formula:

$$(i^*, k^*, j^*) = \arg\max G(i, k, j)\ (i, j \in r) \tag{12}$$

where the city k^* is put into the location of city i^* and city j^*.

3.4 The procedure of VNSHA algorithm

In figure 4, steps 2 and 3 initialize parameters and initial solutions of the algorithm, where the tour cost of insertion position is as max as possible, steps 7 to 8 are to carry out the variable neighborhood search and hill-climbing algorithm based on Wiener process.

The complexity analysis of VNSHA: for MMSTSP problem, the number of salesman m and the number of city n determine the size of problem. The time complexity of initialization, the outer loop of the algorithm, variable neighborhood search, and hill-climbing algorithm is respectively $O(n)$, $O(mn)$, $O(n)$ and $O(mn^{\wedge}2)$. If the generation number is k, P is proportional to city number n. In figure 4, the time complexity of step 2 to 5 is $O(Pmn)$, the time complexity of step 6-8 is $O(Pmn^2)$. The time complexity of VNSHA is $O(kmn^3)$.

The convergence analysis of VNSHA can be discussed by using Markov model [30]. The search space of MMSTSP is I, this algorithm includes VNS and hill-climbing strategy.

$\{A(t), t=0, 1, 2, ..\}$ stands for a population sequence, if $\lim P\{A(t) \cap I^* \neq \phi\}=1 (t \to \infty$, I^* means the set for global optimum), this sequence can converge to global optimum .

P represents a stochastic matrix, C is a primitive matrix, R, $Q \neq 0$, P^k can converge to a matrix, it is defined as $P^{\infty}=\lim(P_{ij})^k=(\pi \ldots \pi)_{T \times T}$ $(k \to \infty)$, where $\pi=(\pi_1, \ldots, \pi_m, 0, \ldots, 0)$, $\pi_j \neq 0$, $1 \leq j \leq m < T$.

$A(t)=\{a_i(t), i \in [1, N], a_i(t) \in I\}$ stands for the population of the generation t, $a_i(t)$ is the candidate solution of the space I, I^* is a set for global optimum, $a^*(t)$ is the best individual, $I^*(t)$ is a set for

$a^*(t)$. $a^*(t)$ is changing randomly over time, when t is ∞, the convergence of $a^*(t)$ means the global convergence of VNSHA. If $P(\lim a^*(t) \in I^*)=1 \Leftrightarrow P(a^* \in \lim A(t))=1(t \to \infty)$, VNSHA is considered to converge. Therefore, this evolution of $a^*(t)$ is a Markov Chain. The steps are shown in figure 4:

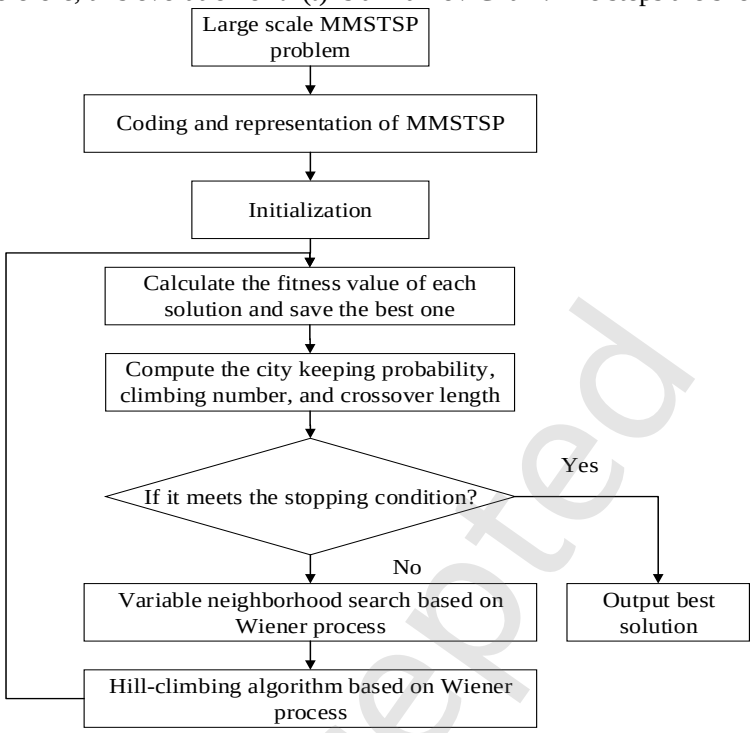

Fig.4. Main steps of VNSHA

3.5 The factors and operators of VNSHA

(1) The factors of VNSHA

According to Wiener process (Brownian motion), the motion of particles are affected by particles radius and environment temperature, the smaller particle radius and the higher temperature mean the stronger movement of the particles. For VNSHA, the solutions by genes coding can be considered to be particles, the activity intensity of the particles are controlled by particles radius and environment temperature, the three factors particles radius, environment temperature and activity intensity based on ITÖ process (Wiener process) for this algorithm can be described as follows.

The particles radius can be defined by the following formula.

$$r(i) = r_{\min} + (r_{\max} - r_{\min})(x - i)/(x - 1) \tag{13}$$

where $r(i)$ stands for the particle radius, x means the number of particles, r_{max} and r_{min} represent respectively the maximum radius and minimum radius, r_{max} is set as 1 and r_{min} is 0 by default.

The environment temperature of this algorithm is computed by the formula.

$$T_i = \eta \cdot T_i \tag{14}$$

where T_i stands for the temperature, η means the coefficient of annealing, it can be set 0.9.

Activity intensity based on ITÖ process are defined as follows.

$$a_i = (e^{-\lambda r_i} - e^{-\lambda r_{\max}})/(e^{-\lambda r_{\min}} - e^{-\lambda r_{\max}}) \times e^{-\frac{1}{T}} \tag{15}$$

where a_i is the activity intensity of the feasible solutions, if the solutions are considered to be particles, the r_i is the particle radius (fitness value) of the particle (solution) i, T is environment environment, r_{max} and r_{min} are set as 1 and 0 by default.

The activity intensity for variable neighborhood solution can be the drift intensity of ITÖ process, and the activity intensity for hill-climbing algorithm is considered to be the interference intensity of ITÖ process (shown in the formula 10). In this paper, the drift intensity is equal to the interference intensity by using the ITÖ process.

(2) Variable neighborhood solution

In figure 5, the cities are reclassified by the city keeping probability p, if $r < P_{cp}$, the cities are kept, else put the cities to the unassigned cities set, reinserting the unassigned cities is based on the

optimization:1) the cities of exclusive set can be only inserted into appointed tour; 2) cities are inserted with the cost as max as possible.

During this process of reinsertion, a constructive heuristic method is used for optimization, the cities are repeatedly inserted based on a distance (cost) measure, which is defined as

$$\Delta_i = \Delta H(i, k, r) = d_{ik} + d_{k(i+1)} - d_{i(i+1)}. \quad (16)$$

If the city i is inserted to the location k, or $\Delta H(i, k, r) = \infty$. In order to compute the best insertion location, the formula is given as follow

$$(i^*, k^*, r^*) = \arg\max H(i, k, r). \quad (17)$$

The city i^* is inserted into route r^* where the distance of location k^* is maximum. The operator is not stopping until no more cities are needed to be inserted.

The city keeping probability P_{cp} based on ITÖ process (Wiener process) are defined as follows.

$$P_{cp} = 1 - \zeta a_i \quad (18)$$

where ζ is random number with uniform distribution, its value is during 0 to 1.

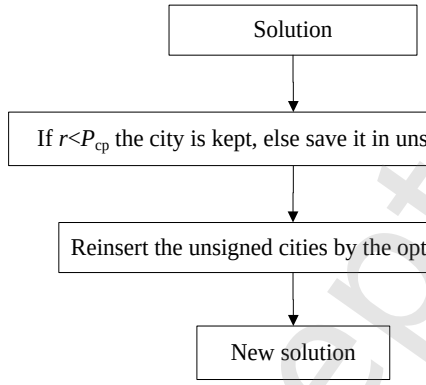

Fig.5. The steps of variable neighborhood solution

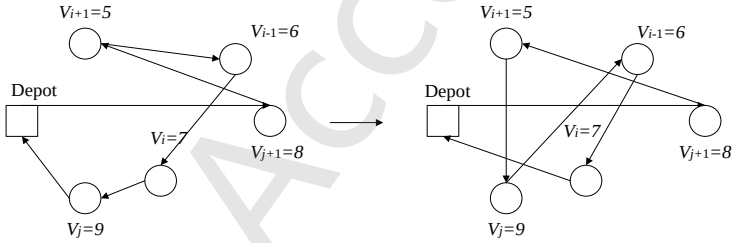

Fig.6. An optimization example of multiple edges

In figure 6, the sequence in the left small figure is 0-8-5-6-7-9-0, the new visiting city route is 0-8-5-9-6-7-0 in right small figure. By variable neighborhood search, the route can be optimized.

Figure 7 is an example of interchange, the sequence in the left small figure is 0-V_1-V_2-V_3-V_4-V_5-V_6, and the city V_2 is interchanged with V_5, the new visiting city route is 0-V_1-V_5-V_3-V_4-V_2-V_6-0.

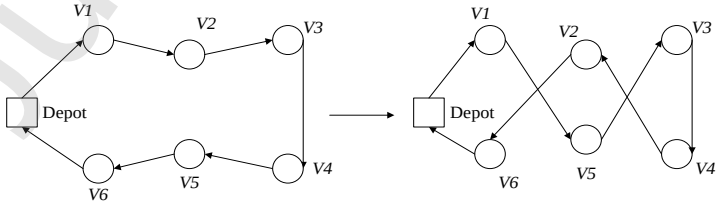

Fig.7. An example of interchange

(3) Hill-climbing algorithm (neighborhood search and optimization)

The extended hill-climbing algorithm (neighborhood search and optimization) belongs to a kind of artificial intelligence algorithm. The advantage of this algorithm is to select some nodes by heuristic strategy for improving efficiency of algorithm.

There are two hill-climbing strategies for optimization, one is interior hill-climbing optimization of current solution by using cities (genes) swapping, another is external hill-climbing optimization of solutions by current solution crossover with the best solution.

1) Interior hill-climbing optimization (Interior neighborhood search and optimization)

For improving the quality of solution, the interior hill-climbing optimization can be carried out

by the neighborhood optimization strategy, which is affected by the neighborhood point selection method, and the step can use two-point swapping for optimization. The climbing number of the cities (genes) can be defined as follows.

$$C_i = \lceil a_i \cdot n/2 \rceil \tag{19}$$

where C is the climbing number of the cities, a_i is the activity intensity of solutions (particles) based on Wiener process, n stands for the number of city, by default C can be set as 1, n/2 or n/4.

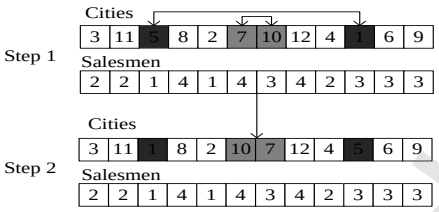

Fig.8. An example of interior hill-climbing optimization

Figure 8 is an example of the hill-climbing operator, in the first step 1, the city 5 is exchanged with city 1, the city 7 is swapped with the city 10, the solution can be optimized by the operator.

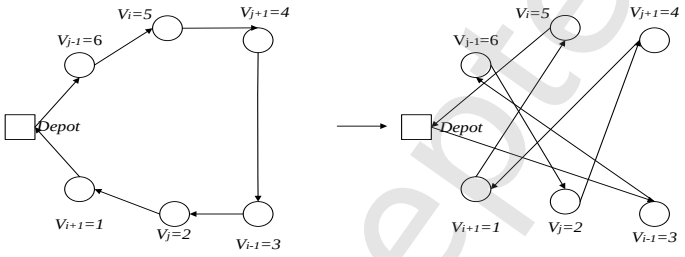

Fig.9. The optimization example of multiple edges

Figure 9 is an optimization example of visiting cities by hill-climbing, the visiting cities are exchanged with multiple cities, by this operator the solution is optimized.

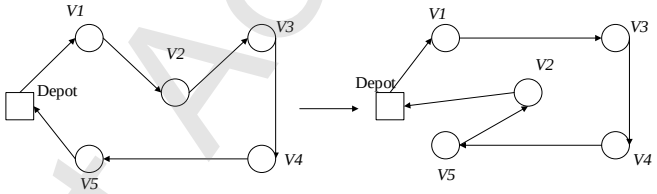

Fig.10. The optimization example of relocation of a single route

Figure 10 stands for the optimization example of relocation for a single route, the access sequence of left figure is 0-V_1-V_2-V_3-V_4-V_5-0, the visiting cities are optimized by the relocation, the new visiting route is 0-V_1-V_3-V_4-V_5-V_2-0.

2) External hill-climbing optimization (External neighborhood search and optimization)

The external hill-climbing optimization can be performed by the current solution crossover with the best solution [42]. After this operator, the solutions can be optimized. The crossover length of crossover operator based on Wiener process is defined as follows.

$$L_i = a_i \times n \tag{20}$$

where L is the length of crossover for current solution with the best solution, n is the city number.

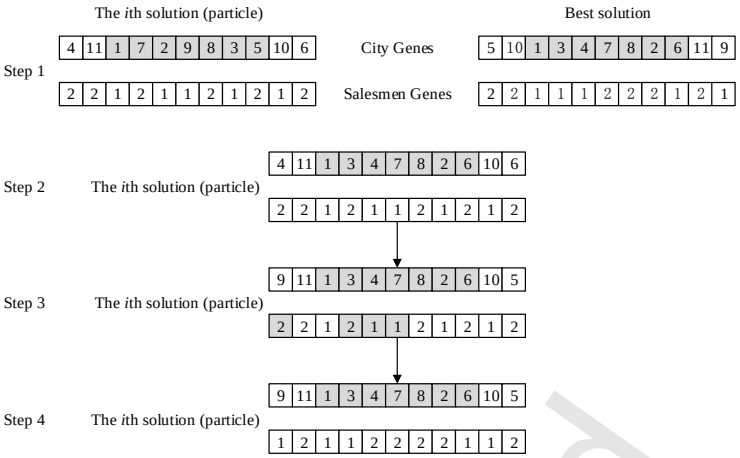

Fig.11. An example of external hill-climbing optimization (crossover operator)

Figure 11 is an example of external hill-climbing (improved crossover operator), the step 1 to step 4 are given in the step 3 to step 6 of the algorithm 5.

The improved crossover operator (adaptive crossover operator) can be carried out. During the crossover process, the solutions can be crossed with the best solution, so that these solutions are improved after certain iterations, the crossover length (crossover rate) is computed by the formula 20, and after the crossover operator the solutions can be further optimized.

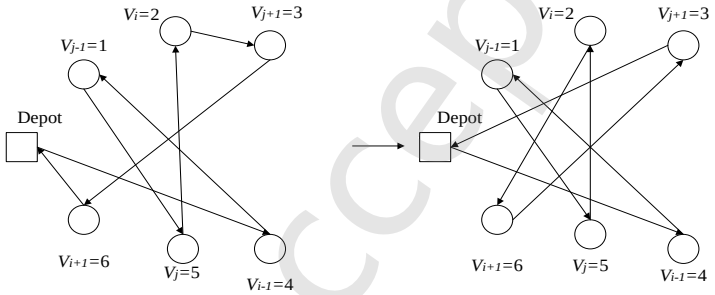

Fig.12. An optimization example of visiting cities for the improved crossover operator

Figure 12 is an optimization example of visiting cities, by this improved crossover operator the visiting city 2 is exchanged. Based on this operator, the solution can be optimized.

4 Experiments and analysis

4.1 Experiment instances

The steps for generating large-scale MMSTSP instances are as follows: for example, if the TSP city number is 7397, then the first city in the TSP city list is divided into 60 sections one by one. Each section has 100 cities, which is the exclusive city set, the remaining cities belong to the last city set (the 61_{th} set), and the last city is usually set as the depot city. Based on MMSTSP instances with n = 7397 and m = 61, if it deletes the last 1000 cities of the 61_{th} city set, it can be converted to an instance of n=6397 and m=61; if it continues to delete the last 1000 cities again, it is converted to an instance of n=5397 and m=61 by performing the same steps as the first instance. Other instances of large scale MMSTSP can be generated by using these steps. It can also refer to the Ref. [10] about the related works of generating large-scale TSP versions instances.

Table 2 gives the MMSTSP instances with city number less than 1000.

Table 2 The MMSTSP instances

Instance	***n***	***m***	***e***	***s***
Small				
1	21	2	10	1
2	31	2	15	1
3	31	3	10	1
4	41	2	20	1
5	41	3	13,14	1
6	41	4	10	1
7	51	3	16,17	1
8	51	4	12,13	1
9	51	5	10	1

10	76	3	25	1
11	76	4	18,19	1
12	76	5	15	1
13	76	6	12,13	1
Medium				
14	101	4	25	1
15	101	5	20,18,21	1
16	101	6	16,17	1
17	101	7	14,15	1
18	206	9	22,23	1
19	431	12	35,36	1
20	547	14	39	1
21	612	23	26,27	1
22	665	33	20,21	1

In table 2 and table 3, *n* is the number of city, *m* stands for the number of salesperson, *e* is the exclusive city set, *s* represents the starting and ending point.

The following table 3 is the large scale MMSTSP instances of the experiments.

Table 3 Large scale MMSTSP instances

Instance	***n***	***m***	***e***	***s***
Large				
1	1397	12	100,278	1
2	1397	22	50,100,278	1
3	1397	42	25,100,278	1
4	2461	5	600,60	1
5	2461	8	300,360	1
6	2461	32	60,600	1
7	2461	42	45,60,600	1
8	3461	13	250, 460	1
9	3461	25	125,460	1
10	3461	41	75,460	1
11	5397	41	100,1396	1
12	5397	61	80,596	1
13	6397	41	150,396	1
14	6397	51	120,396	1
15	6397	61	100,396	1
16	7397	41	150,1396	1
17	7397	51	120,1396	1
18	7397	61	100,1396	1

4.2 Parameters and sensitivity analysis

The experiments are done in the computer environment of the Ref. [17]. All the algorithms perform same time in the experiments as the stopping condition.

The settings of parameters for HCGA, GA, GAG, and SAGA are the same with the Ref. [1, 17]; the improved hill-climbing algorithm (IHA) [17]: the population is 160; VNSHA: population number is 160. The reason for this setting appropriately is that the algorithms can show better performance by the settings of parameters based on lots of experiments and data analysis.

For VNSHA: the city keeping probability P_{cp}, climbing algebra C and crossover length L are computed by the given formulas. The experiments are made to analyze the parameter population of VNSHA and its sensitivity analysis.

Table 4. Solution quality of three instances of VNSHA with different population number

Population	***n*=2461, *m*=5**		***n*=5397, *m*=41**		***n*=7397, *m*=41**	
Number	**Best**	**Mean**	**Best**	**Mean**	**Best**	**Mean**
20	194.0	179.9	2594.0	2351.9	2114.0	1575.1
40	196.0	181.2	2746.0	2375.2	1771.0	1623.0
60	187.0	178.7	2740.0	2487.5	2190.0	1783.9
80	190.0	175.3	2746.0	2473.6	1901.0	1747.4
100	192.0	180.7	2746.0	2580.0	2043.0	1775.2
120	202.0	182.4	2594.0	2488.3	2107.0	1812.3
140	191.0	183.7	2763.0	2520.7	2115.0	1817.1
160	**202.0**	**186.4**	**2732.0**	**2569.8**	**2329.0**	**1867.7**
180	202.0	185.6	2732.0	2542.4	2012.0	1809.0
200	202.0	184.3	2577.0	2478.1	2043.0	1880.9

For the table 4, it is the solution quality of the proposed algorithm with three instances and different population number, the three instances are n=2461 m=5, n=5397 m=41, and n=7397 m=41, the best and mean are the best solution and average solution of the algorithms running ten

times for this problem. From the table 4, it can show that the best and mean solutions are similar with the different number of population, according to the solution quality of VNSHA, there is small sensitivity for the different number of population.

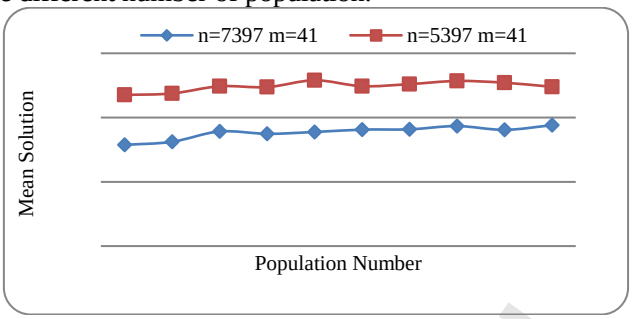

Fig.13. The mean solution quality of VNSHA with the different number of population

Figure 13 is the mean solution of the proposed algorithm of the two instances (*n*=7397 and *n*=5397) with the different population number.

The table and figure show that when the population number is 160, VNSHA can display relatively better best and mean solution quality for large scale optimization of MMSTSP.

4.3 Experiment running interfaces

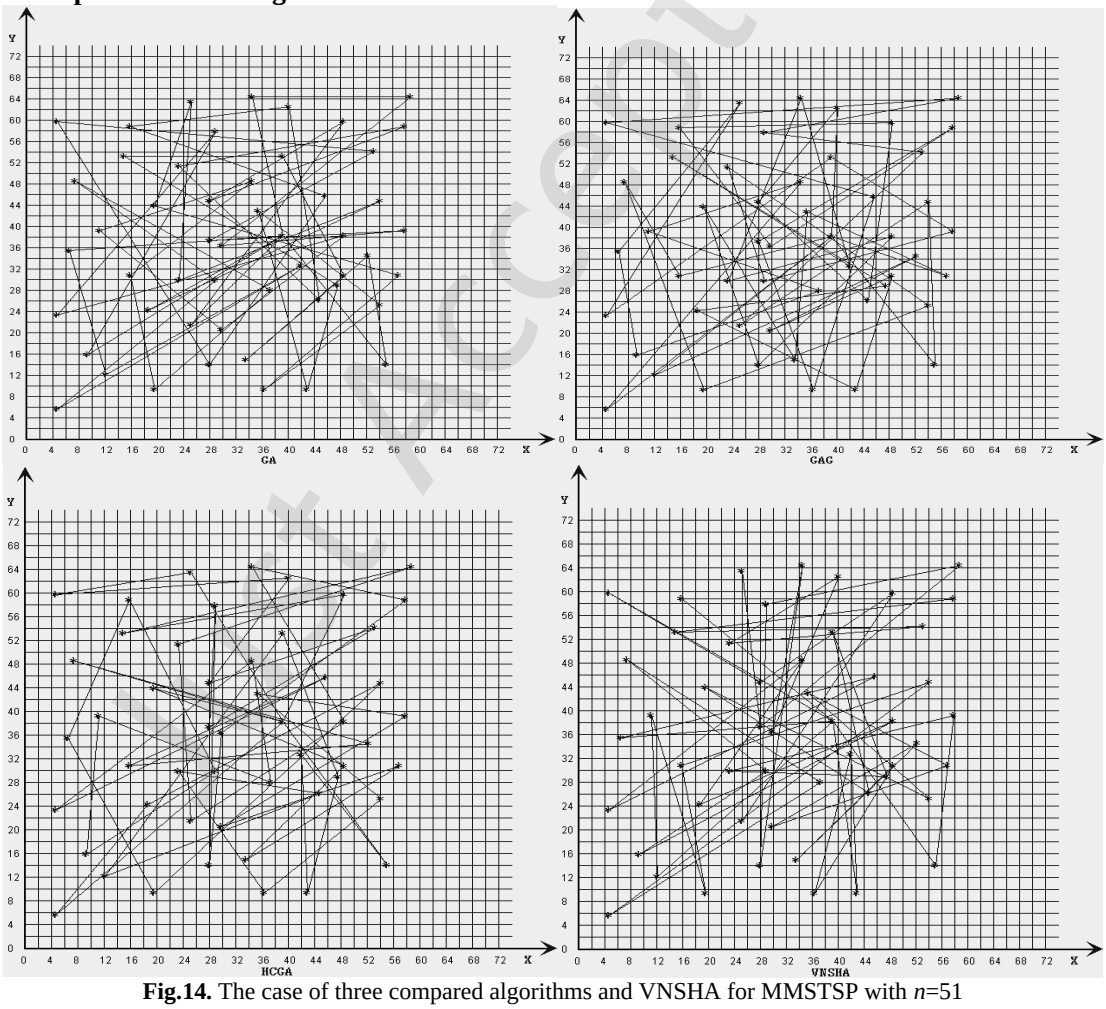

Fig.14. The case of three compared algorithms and VNSHA for MMSTSP with *n*=51

In the figure 14, GA: the mean solution as 18.0, iteration as 33226; GAG: mean solution is 18.0, iteration is 50437; HCGA: mean solution as 19.6, iteration as 44841; VNSHA: mean solving solution is 21.4, iteration is 4966.

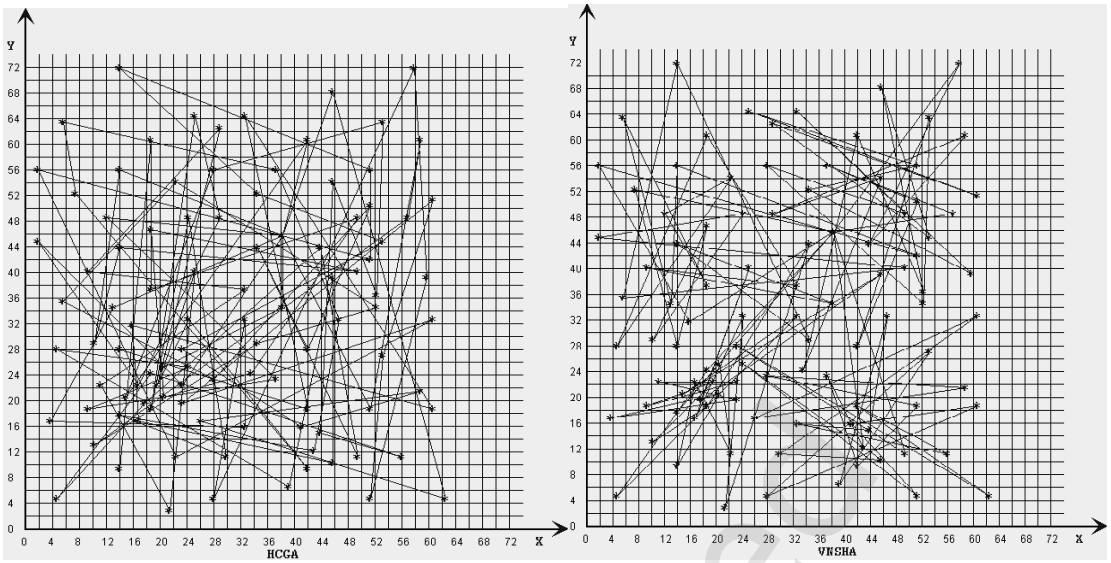

Fig.15. The case of the compared algorithm and VNSHA for MMSTSP with n=101

In figure 15, HCGA: mean solution 14.4, iteration 17033; VNSHA: mean solving solution as 22.2, iteration is 3096.

Figures 14-15 are used as illustrative examples of the running interface or routes, with quantitative performance superiority already established by the statistical results in the tables.

The visual interfaces are given by using the small scale MMSTSP instances, the experiment results show that VNSHA has better performance than GA, GAG and HCGA in term of mean solution quality.

In the figures 14-15, they try to choose the longest path from one point to another, which is consistent with the objective function of MMSTSP, it can lead to this situation of the simulation interfaces that visiting paths are interlaced by using these instances.

4.4 Experiments for large scale MMSTSP

The following tables are experiment results for MMSTSP by using large scale data. In the tables 6-7, "mean" and "best" respectively stand for the mean (average) solution and best solution, iteration is iterations number of the algorithm for this problem.

The solution quality (such as best solution or mean solution) is under equal running time in the experimental setup, as per-iteration computational costs vary across algorithms, and iteration counts are not the efficiency metrics.

HCGA, SAGA, GA, GAG are genetic algorithm and improved genetic algorithms from Ref. [1, 27]; IHA is an improved hill-climbing algorithm with local search [17]; HA is hill-climbing algorithm (HA); VNS-1 is a hybrid ITÖ algorithm based on VNS and ITÖ algorithm [41]; VNSHA is the hybrid variable neighborhood search, which is proposed in this paper.

The table 5 displays the related information for the experiments such as the number of repeated runs, stopping condition, and the definition of the iterations.

Table 5 The experimental protocol summary table

Factors	**Explanation**
The number of repeated runs	The running times 10
Stopping criteria	Same running time (such as 60s or 180s)
Reported statistics	Best solution, mean solution and iteration
The definition of key metrics such as iterations	The total number of rounds or generations during the solving process

In table 5, there are some factors, the number of repeated runs is 10 where the algorithms run 10 times for solving problem; the stopping condition or criteria is that the algorithms run same running time, in the experiments the algorithms run 60s for solving the problem in tables 6-8, the algorithms run 180s in tables 9-11; best solution, mean solution and iteration are the reported statistics; the definition of iterations is the number of iterations of the algorithms refers to the total number of rounds or generations in which the algorithms repeatedly perform core operations (such as updating solutions, evaluating fitness, etc.) during the solving process.

Table 6 Experiment results of the three algorithms with running 60s (Unit km)

Instance	n	m	GA			GAG			HCGA		
			Best	**Mean**	**Iteration**	**Best**	**Mean**	**Iteration**	**Best**	**Mean**	**Iteration**
1	1397	12	45.0	40.6	1732	44.0	40.8	1389	41.0	38.4	1293
2	1397	22	39.0	34.3	1132	37.0	33.9	1599	39.0	32.8	1035
3	1397	42	30.0	28.3	527	34.0	29.3	816	33.0	29.5	922
4	2461	5	71.0	53.9	520	67.0	54.7	742	65.0	59.8	677
5	2461	8	51.0	45.3	573	50.0	47.0	432	50.0	44.0	422
6	2461	32	32.0	30.9	242	36.0	31.3	441	36.0	31.2	331
7	2461	42	31.0	29.5	291	36.0	30.6	357	31.0	29.7	184
8	3461	13	42.0	35.8	245	42.0	37.4	281	42.0	35.4	255
9	3461	25	32.0	29.8	192	31.0	30.4	315	32.0	30.4	267
10	3461	41	32.0	28.6	144	31.0	27.3	208	31.0	25.6	196
11	5397	41	488.0	430.4	200	457.0	374.0	192	463.0	410.9	197
12	5397	61	478.0	372.5	153	385.0	339.7	181	400.0	320.7	145
13	6397	41	442.0	328.2	119	330.0	290.8	95	366.0	285.1	97
14	6397	51	381.0	305.2	100	296.0	255.6	99	314.0	239.5	85
15	6397	61	303.0	272.6	118	272.0	223.6	98	291.0	241.9	89
16	7397	41	341.0	305.5	98	253.0	219.7	50	295.0	251.3	61
17	7397	51	323.0	270.0	66	306.0	231.9	48	331.0	241.0	65
18	7397	61	322.0	247.2	74	262.0	208.3	49	249.0	210.7	66
	Standard	**deviations**	177.5	144.5	419.8	145.6	121.4	441.0	154.5	125.7	363.4

Table 7 Experiment results of the two algorithms and VNSHA with running 60s (Unit km)

Instance	n	m	SAGA			IHA			VNSHA		
			Best	**Mean**	**Iteration**	**Best**	**Mean**	**Iteration**	**Best**	**Mean**	**Iteration**
1	1397	12	41.0	37.1	579	85.0	81.3	1090	105.0	96.3	85
2	1397	22	40.0	31.8	443	79.0	70.6	913	90.0	85.6	90
3	1397	42	35.0	28.4	385	57.0	52.2	681	72.0	65.4	87
4	2461	5	52.0	47.8	329	196.0	182.2	590	213.0	200.2	16
5	2461	8	50.0	42.1	276	157.0	142.2	610	156.0	152.6	25
6	2461	32	31.0	29.6	202	93.0	81.8	431	113.0	101.3	28
7	2461	42	31.0	27.6	170	79.0	66.9	375	91.0	81.3	28
8	3461	13	41.0	33.6	207	129.0	117.5	419	135.0	127.9	17
9	3461	25	31.0	29.3	162	114.0	98.9	361	118.0	109.0	19
10	3461	41	30.0	26.4	122	92.0	86.6	292	105.0	99.6	20
11	5397	41	497.0	372.0	149	2817.0	2551.3	154	2882.0	2763.0	4
12	5397	61	350.0	292.2	123	2062.0	1882.3	133	2246.0	2086.6	9
13	6397	41	338.0	267.8	106	2196.0	1798.6	113	2231.0	1983.8	7
14	6397	51	313.0	243.1	113	2074.0	1839.9	91	2307.0	2093.6	7
15	6397	61	249.0	219.5	91	1313.0	914.6	83	1295.0	1175.1	7
16	7397	41	315.0	254.8	45	2166.0	1837.1	36	2260.0	2002.7	3
17	7397	51	293.0	239.5	60	2256.0	1911.6	36	2292.0	1981.8	3
18	7397	61	232.0	213.5	71	1204.0	949.3	34	1220.0	1073.9	3
	Standard	**deviations**	150.5	118.4	142.3	1000.6	869.2	306.4	**1040.0**	**948.5**	**28.8**

In tables 6 to 7, it shows that VNSHA demonstrates better means solution and best solution than the GA, GAG, HCGA, SAGA, IHA, it also shows less iterations than them.

In table 8, VNSHA has the biggest average value of the best percentage deviation and mean percentage deviation, it can display better solution than IHA, HCGA, GA, GAG, and SAGA.

Table 8 Percentage deviation of the algorithms with running 60s (Unit km)

Instance	n	m	GA		GAG		HCGA		SAGA		IHA		VNSHA	
			PD_{best}	PD_{av}	PD_{best}	PD_{av}	PD_{best}	PD_{av}	PD_{best}	PD_{av}	PD_{best}	PD_{av}	PD_{best}	PD_{av}
1	1397	12	9.7	9.4	7.3	9.9	0	3.5	0	0	107.3	119.1	156.0	159.5
2	1397	22	5.4	7.8	0	6.6	5.4	3.1	8.1	0	113.5	122.0	143.2	169.1
3	1397	42	0	0	13.3	3.5	10	4.2	16.6	0.3	90	84.4	140	131.0
4	2461	5	36.5	12.7	28.8	14.4	25	25.1	0	0	276.9	281.1	309.6	318.8
5	2461	8	2	7.6	0	11.6	0	4.5	0	0	214	237.7	212	262.4
6	2461	32	3.2	4.3	16.1	5.7	16.1	5.4	0	0	200	176.3	264.5	242.2
7	2461	42	0	6.8	16.1	10.8	0	7.6	0	0	154.8	142.3	193.5	194.5
8	3461	13	2.43	6.5	2.4	11.3	2.4	5.	0	0	214.6	249.7	229.2	280.6
9	3461	25	3.2	1.7	0	3.7	3.2	3.7	0	0	267.7	237.5	280.6	272.0
10	3461	41	6.6	11.7	3.3	6.6	3.3	0	0	3.1	206.6	238.2	250	289.0
11	5397	41	6.7	15.6	0	0.5	1.3	10.4	8.7	0	516.4	585.8	530.6	642.7
12	5397	61	36.5	27.4	10	16.2	14.2	9.7	0	0	489.1	544.1	541.7	614.0
13	6397	41	33.9	22.5	0	8.5	10.9	6.4	2.4	0	565.4	571.6	576.0	640.7
14	6397	51	28.7	27.4	0	6.7	6.0	0	5.7	1.5	600.6	668.2	679.3	774.1
15	6397	61	21.6	24.1	9.2	1.8	16.8	10.2	0	0	427.3	316.6	420.0	435.3
16	7397	41	34.7	39.0	0	0	16.6	14.3	24.5	15.9	756.1	736.1	793.2	811.5
17	7397	51	10.2	16.4	4.4	0	12.9	3.9	0	3.2	669.9	724.3	682.2	754.5
18	7397	61	38.7	18.6	12.9	0	7.3	1.1	0	2.4	418.9	355.7	425.8	415.5
Average			15.5	14.4	6.8	6.5	8.4	6.6	3.6	1.4	349.4	355.0	**379.3**	**411.5**

Table 9 Experiment results of the three algorithms with running 180s (Unit: km)

Instance	n	m	GA			GAG			HCGA		
			Best	Mean	Iteration	Best	Mean	Iteration	Best	Mean	Iteration
1	1397	12	68.0	49.8	8207	48.0	43.2	4674	83.0	51.0	4791
2	1397	22	58.0	38.7	4660	42.0	35.9	4537	52.0	38.5	5544
3	1397	42	33.0	31.2	4455	34.0	30.8	2211	37.0	31.2	2505
4	2461	5	68.0	62.3	3005	79.0	62.9	2057	86.0	63.5	2138
5	2461	8	55.0	49.9	1816	74.0	52.1	2508	63.0	53.5	1741
6	2461	32	36.0	31.7	1290	36.0	30.9	508	46.0	34.8	1164
7	2461	42	40.0	33.6	1730	36.0	31.2	954	32.0	31.1	1183
8	3461	13	43.0	40.0	1091	51.0	41.9	1107	45.0	38.3	747
9	3461	25	36.0	31.9	639	36.0	31.4	665	36.0	31.8	627
10	3461	41	32.0	29.8	465	31.0	29.3	628	32.0	29.0	521
11	5397	41	649.0	519.4	598	593.0	481.4	577	630.0	556.9	516
12	5397	61	464.0	421.6	565	494.0	409.7	501	522.0	444.2	538
13	6397	41	475.0	360.5	360	415.0	367.5	382	484.0	374.1	331
14	6397	51	401.0	334.4	365	413.0	329.8	366	393.0	331.9	282
15	6397	61	351.0	294.0	319	398.0	313.8	415	353.0	299.7	264
16	7397	41	432.0	350.2	226	394.0	324.4	335	390.0	332.7	253
17	7397	51	390.0	333.3	285	417.0	300.5	227	368.0	309.0	252
18	7397	61	383.0	307.7	298	308.0	272.3	305	357.0	286.9	201

Table 10 Experiment results of SAGA, IHA and VNSHA with running 180s (Unit km)

Instance	n	m	SAGA			IHA			VNSHA		
			Best	Mean	Iteration	Best	Mean	Iteration	Best	Mean	Iteration
1	1397	12	42.0	39.6	2055	85.0	81.6	3163	119.0	109.6	218
2	1397	22	39.0	33.3	1256	79.0	72.3	2609	109.0	97.2	229
3	1397	42	33.0	29.8	1336	63.0	54.6	1917	96.0	80.1	219
4	2461	5	61.0	56.7	1067	202.0	187.2	1684	212.0	206.9	47
5	2461	8	52.0	45.1	1150	143.0	134.8	1655	166.0	157.9	67
6	2461	32	36.0	31.9	620	93.0	79.6	1207	140.0	112.5	73
7	2461	42	32.0	29.7	629	72.0	63.0	1059	143.0	98.7	72
8	3461	13	41.0	35.8	476	123.0	115.2	1172	158.0	145.7	52
9	3461	25	32.0	30.1	402	109.0	100.3	1001	131.0	122.6	59
10	3461	41	31.0	28.3	480	100.0	88.4	823	131.0	113.8	60
11	5397	41	634.0	505.3	419	2740.0	2548.0	489	3208.0	2961.1	15
12	5397	61	452.0	373.9	336	2060.0	1903.3	402	2356.0	2291.4	27
13	6397	41	1073.0	388.2	255	2064.0	1801.4	294	2594.0	2231.8	20
14	6397	51	356.0	297.1	248	1982.0	1852.5	267	2299.0	2193.6	20
15	6397	61	296.0	275.1	238	1339.0	976.4	244	1559.0	1414.0	20
16	7397	41	359.0	303.3	199	2301.0	1920.4	127	2354.0	2134.2	9
17	7397	51	344.0	292.2	190	2209.0	1871.7	121	2299.0	2106.9	9
18	7397	61	274.0	241.5	163	1271.0	955.5	116	1390.0	1264.7	9

From table 9 to table 10, VNSHA can demonstrate better performance than HCGA, SAGA, GA, GAG, and IHA on mean solution and best solution.

Table 11 Percentage deviation of the algorithms by running 180s (Unit km)

Instance	n	m	GA		GAG		HCGA		SAGA		IHA		VNSHA	
			PD_{best}	PD_{av}	PD_{best}	PD_{av}	PD_{best}	PD_{av}	PD_{best}	PD_{av}	PD_{best}	PD_{av}	PD_{best}	PD_{av}
1	1397	12	61.9	25.7	14.2	9.0	97.6	28.7	0	0	102.3	106.0	183.3	176.7
2	1397	22	48.7	16.2	7.6	7.8	33.3	15.6	0	0	102.5	117.1	179.4	191.8
3	1397	42	0	4.6	3.0	3.3	12.1	4.6	0	0	90.9	83.2	190.9	168.7
4	2461	5	11.4	9.8	29.5	10.9	40.9	11.9	0	0	231.1	230.1	247.5	264.9
5	2461	8	5.7	10.6	42.3	15.5	21.1	18.6	0	0	175	198.8	219.2	250.1
6	2461	32	0	2.5	0	0	27.7	12.6	0	3.2	158.3	157.6	288.8	264.0
7	2461	42	25	13.1	12.5	5.0	0	4.7	0	0	125	112.1	346.8	232.3
8	3461	13	4.8	11.7	24.3	17.0	9.7	6.9	0	0	200	221.7	285.3	306.9
9	3461	25	12.5	5.9	12.5	4.3	12.5	5.6	0	0	240.6	233.2	309.3	307.3
10	3461	41	3.2	5.3	0	3.5	3.2	2.4	0	0	222.5	212.3	322.5	302.1
11	5397	41	9.4	7.8	0	0	6.2	15.6	6.9	4.9	362.0	429.2	440.9	515.1
12	5397	61	2.6	12.7	9.2	9.5	15.4	18.8	0	0	355.7	409.0	421.2	512.8
13	6397	41	14.4	0	0	1.9	16.6	3.7	158.5	7.6	397.3	399.6	525.0	519.0
14	6397	51	12.6	12.5	16.0	11.0	10.3	11.7	0	0	456.7	523.5	545.7	638.3
15	6397	61	18.5	6.8	34.4	14.0	19.2	8.9	0	0	352.3	254.9	426.6	413.9
16	7397	41	20.3	15.4	9.7	6.9	8.6	9.6	0	0	540.9	533.1	555.7	603.6
17	7397	51	13.3	14.0	21.2	2.8	6.9	5.7	0	0	542.1	540.5	568.3	621.0
18	7397	61	39.7	27.4	12.4	12.7	30.2	18.7	0	0	363.8	295.6	407.2	423.6
Average			16.9	11.2	13.8	7.5	20.6	11.4	9.1	0.8	278.8	281.0	**359.1**	**372.9**

Table 12 Experiment results of the HA and VNSHA algorithms (Unit: km)

Instance	n	m	HA		VNSHA	
			Best	**Mean**	**Best**	**Mean**
1	1397	12	91.0	67.8	109.0	101.5
2	1397	22	69.0	53.2	93.0	87.4
3	1397	42	56.0	42.2	73.0	69.2
4	2461	5	572.0	527.8	201.0	190.7
5	2461	8	418.0	158.2	168.0	152.4
6	2461	32	42.0	38.5	101.0	91.3
7	2461	42	42.0	35.6	84.0	76.7
8	3461	13	82.0	63.0	136.0	123.0
9	3461	25	61.0	50.6	121.0	107.8
10	3461	41	50.0	39.1	103.0	97.5
11	5397	41	786.0	642.3	2837.0	2555.2
12	5397	61	1031.0	629.0	2068.0	1842.5
13	6397	41	1065.0	842.6	2074.0	1910.4
14	6397	51	773.0	672.8	2124.0	1852.9
15	6397	61	952.0	589.6	1117.0	925.9
16	7397	41	1145.0	752.7	2033.0	1772.9
17	7397	51	821.0	645.5	1970.0	1769.9
18	7397	61	701.0	596.6	1281.0	986.6
Standard	**deviations**		412.9	304.7	**963.6**	**856.9**

In tables 8 and 11, the best and average percentage deviations are computed by formulas 21-22:

$$PD_{best} = \frac{best\ known\ solution - best\ solution}{best\ known\ solution} \times 100 \quad (21)$$

$$PD_{av} = \frac{best\ known\ solution - average\ solution}{best\ known\ solution} \times 100 \quad (22)$$

Table 13 Experiment results of VNS, VNS-1 and VNSHA (Unit km)

Instance	n	m	VNS		VNS-1		VNSHA	
			Best	**Mean**	**Best**	**Mean**	**Best**	**Mean**
1	1397	12	57.0	50.7	56.0	56.0	109.0	101.5
2	1397	22	45.0	41.1	51.0	41.6	93.0	87.4
3	1397	42	38.0	34.3	38.0	33.8	73.0	69.2
4	2461	5	545.0	456.0	545.0	462.3	201.0	190.7
5	2461	8	279.0	127.6	369.0	121.5	168.0	152.4
6	2461	32	40.0	35.8	41.0	37.0	101.0	91.3
7	2461	42	42.0	34.7	37.0	33.8	84.0	76.7
8	3461	13	64.0	58.2	68.0	60.7	136.0	123.0
9	3461	25	57.0	46.7	51.0	45.1	121.0	107.8
10	3461	41	40.0	35.8	41.0	36.2	103.0	97.5
11	5397	41	796.0	630.7	775.0	644.5	2837.0	2555.2
12	5397	61	628.0	534.3	599.0	555.1	2068.0	1842.5
13	6397	41	838.0	705.5	1041.0	769.1	2074.0	1910.4
14	6397	51	696.0	621.5	871.0	673.7	2124.0	1852.9
15	6397	61	631.0	534.1	984.0	570.0	1117.0	925.9
16	7397	41	1063.0	796.8	1055.0	798.1	2033.0	1772.9
17	7397	51	688.0	621.1	871.0	671.4	1970.0	1769.9
18	7397	61	706.0	572.8	742.0	557.3	1281.0	986.6
Standard	**deviations**		348.4	286.8	399.7	300.2	**963.6**	**856.9**

VNSHA shows the biggest deviation than GA, GAG, HCGA, SAGA and IHA in table 11, which means that VNSHA has better performance (solutions) than the compared algorithms.

In the tables 12-14, HA is hill-climbing algorithm of this paper; VNS-1 is a hybrid ITÖ algorithm based on VNS and ITÖ algorithm [41]; VNSHA is proposed algorithm of the paper.

In tables 12-13, the best and mean are the best solution and mean solution of the algorithms running 10 times, and running same time is the stopping condition.

In table 14, the smaller percentage deviation means better solution quality, the best and average percentage deviations are computed by the formulas 23-24:

$$PD_{best} = \frac{best\ solution - worst\ known\ solution}{worst\ known\ solution} \times 100 \quad (23)$$

$$PD_{av} = \frac{average\ solution - worst\ known\ solution}{worst\ known\ solution} \times 100 \quad (24)$$

Table 12 and table 13 show that VNSHA demonstrates better best and mean solutions than the compared algorithms HA, VNS and VNS-1. In table 14, VNSHA has better deviations than the compared algorithms HA, VNS and VNS-1.

Table 14 Percentage deviations of the HA, VNS, VNS-1 and VNSHA algorithms

Instance	n	m	HA		VNS		VNS-1		VNSHA	
			PD_{best}	PD_{av}	PD_{best}	PD_{av}	PD_{best}	PD_{av}	PD_{best}	PD_{av}
1	1397	12	16.5	33.2	47.7	50.0	48.6	44.8	0	0
2	1397	22	25.8	39.1	51.6	52.9	45.1	52.4	0	0
3	1397	42	23.2	39.0	47.9	50.4	47.9	51.1	0	0
4	2461	5	0	0	4.7	13.6	4.7	12.4	64.8	63.8
5	2461	8	0	0	33.2	19.3	11.7	23.1	59.8	3.6
6	2461	32	58.4	57.8	60.3	60.7	59.4	59.4	0	0
7	2461	42	50	53.5	50	54.7	55.9	55.9	0	0
8	3461	13	39.7	48.7	52.9	52.6	50	50.6	0	0
9	3461	25	49.5	53.0	52.8	56.6	57.8	58.1	0	0
10	3461	41	51.4	59.8	61.1	63.2	60.1	62.8	0	0
11	5397	41	72.2	74.8	71.9	75.3	72.6	74.7	0	0
12	5397	61	50.1	65.8	69.6	71.0	71.0	69.8	0	0
13	6397	41	48.6	55.8	59.5	63.0	49.8	59.7	0	0
14	6397	51	63.6	63.6	67.2	66.4	58.9	63.6	0	0
15	6397	61	14.7	36.3	43.5	42.3	11.9	38.4	0	0
16	7397	41	43.6	57.5	47.7	55.0	48.1	54.9	0	0
17	7397	51	58.3	63.5	65.0	64.9	55.7	62.0	0	0
18	7397	61	45.2	39.5	44.8	41.9	42.0	43.5	0	0
Average			39.5	15.5	51.7	11.2	47.3	11.1	**6.9**	**6.6**

There are limitations for the traditional algorithms such as bad convergence and easily falling into local optimum, the proposed algorithm VNSHA can display better performance by improving these limitations. Based on the experiments and data, it shows that VNSHA demonstrates better solution quality than the compared algorithms HA, VNS, VNS-1, GA, GAG, HCGA, and SAGA.

4.5 Discussions

For testing the effectiveness of VNSHA algorithm, this article mainly uses MMSTSP data to do experiments. For the MMSTSP with city number less than 1000, the experiments are to run 60s as termination condition, the results and data show that VNSHA demonstrates better solution quality than GA, GAG, HCGA; for the MMSTSP with city number more than 1000, it utilizes running same time as the stopping condition, the tables 6-14 display that VNSHA can show improvement over HA, VNS, VNS-1, GA, GAG, HCGA, SAGA and IHA in term of solution quality.

MMSTSP is a new combinational optimization problem, which can be used to model some real-world engineering problems. The traditional algorithms for MMSTSP, such as GA and HCGA, are constrained on solution quality and convergence speed. For improving the problems, the research proposes a novel hybrid VNS for solving it, the results and data prove its effectiveness. VNSHA uses variable neighborhood search and hill-climbing algorithm to optimize solutions. Based on the analysis of results, VNSHA shows better solution than HA, VNS, VNS-1, GAG, GA, SAGA, HCGA, and IHA for large scale optimization of MMSTSP problem.

VNSHA is an effective intelligent optimization algorithm, it can advance the state of the art by overcoming the limitations of the existing work such as the bad convergence and easily falling local optimum. The proposed method uses VNS and hill-climbing for optimization, VNS is to improve the solutions by deleting and reinserting cities, hill-climbing algorithm is performed by using interior hill-climbing optimization and external hill-climbing optimization during the solving process, they are carried out by swapping and crossover operator. In addition, the initial operator can also effectively improve the solutions. Based on the mentioned optimization strategies, the proposed algorithm VNSHA can demonstrate better performance than the traditional compared state-of-the-art algorithms on the solution quality.

4.6 Hypothesis test and analysis

Variable input: input variable 1 (the mean solution of VNSHA) and variable 2 (the mean solution of SAGA) data respectively; assumed mean difference: if the two values are equal, enter 0; if the difference between the values is equal to a constant, enter the constant. Significant level: generally 0.1, 0.05 or 0.01, fill in as required; *z*: calculated *z* value.

Samples are taken from two normal distribution with unknown mean and known standard deviation. The first has a standard deviation of 1053.973, a sample size of 18, and a sample mean of 963.0679; in the second population, the standard deviation is 162.1179, the sample size is 18, and the sample mean is 147.4296; the right tail test is conducted when the difference between the mean values of the two populations is equal to 0 at the significance level of 0.01.

According to the meaning of the question, the statistical assumptions are as follows:
H_0: $\mu_1 - \mu_2 = 0$
H_1: $\mu_1 - \mu_2 > 0$
$\alpha = 0.01$
Using the statistic of $\alpha = 0.01$, representing right tail decision rules with P values:
If $P =< 0.01$, then reject H_0;
Compute P value:

$$z = \frac{(x_1 - x_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} = \frac{(963.0679 - 147.4296) - 0}{\sqrt{\frac{1053.973^2}{18} + \frac{162.1179^2}{18}}} = 0.013 \quad (25)$$

Check the normal distribution table, while z=0.013, the P is obtained based on the z. Because P is less than 0.01, thus reject the null hypothesis. Therefore, at 0.01 significant level, the mean difference between the two algorithms VNSHA and SAGA is obviously greater than 0, it show that $\mu_1 > \mu_2$. According to these steps, the hypothesis test of VNSHA with HA, VNS, VNS-1, GAG, GA, HCGA, and IHA are made based on the mean solution quality of the algorithms.

This analysis adopts paired sample comparison methods (paired statistical tests) for VNSHA, VNS-1 and VNS. Normality Test: Shapiro-Wilk test for paired differences in each group; Significance level α=0.05; p-value >0.05 indicates normal distribution. Test Execution: Paired t-test if paired differences are normally distributed; Wilcoxon signed-rank test if paired differences are non-normal. Result Interpretation: p-value <0.05 indicates significant difference between groups; p-value >=0.05 indicates no significant difference.

Table 15 Paired statistical tests for VNSHA, VNS-1 and VNS

Comparison Pair	Test Method	Statistic	p-value	Significance(α=0.05)
VNS vs VNS-1	Wilcoxon signed-rank test	W=98.0	0.901	Not significant
VNS vs VNSHA	Wilcoxon signed-rank test	W=0.0	<0.001	Significant
VNS-1 vs VNSHA	Wilcoxon signed-rank test	W=0.0	<0.001	Significant

In table 15, VNS and VNS-1 have similar performance with no statistical difference; VNSHA has extreme significant differences from VNS and VNS-1, and VNSHA can show improvements and advantages. (1) VNS vs VNS-1. Normality Test: Paired differences failed normality test (p<0.05); Selected Test: Wilcoxon signed-rank test; Result: p-value=0.901; Conclusion: No significant difference between groups, likely minor variants of the same algorithm. (2) VNS vs VNSHA. Normality Test: Non-normal paired differences (p<0.001); Selected Test: Wilcoxon signed-rank test; Result: p-value<0.001; Conclusion: Extreme significant difference between VNSHA and VNS, with VNSHA generally showing higher values. (3) VNS-1 vs VNSHA. Normality Test: Non-normal paired differences (p<0.001); Selected Test: Wilcoxon signed-rank test; Result: p-value<0.001; Conclusion: VNSHA significantly different from VNS-1.

5 Conclusions and next works

The research firstly studies large-scale MMSTSP problem and provides a new VNS to solve it. The results show VNSHA demonstrates not only superiority over traditional algorithms HCGA, SAGA, GAG and GA on solution quality, but also displays better solution than the versions of VNS. For the proposed algorithm, there are the advantages and merits: compared to the traditional intelligent optimization algorithms such GA and its versions, the Wiener process is applied to the proposed algorithm, which can provide the basic theory for it; the operators of VNSHA algorithm can be controlled by Wiener process, which can reduce parameters of the algorithm compared with the traditional algorithms; the improved VNS and extended hill-climbing (including the interior and external optimization) based on Wiener process are used for further optimization of the large scale MMSTSP. However, there are also limitations for the proposed algorithm, the running time of VNSHA is more than the basic algorithms such as GA, and the complexity is relatively large.

The future works can be summarized: 1) in this paper, the scale of MMSTSP is more than 1000, but it is still limited, the next research can study the scale with the city number more than 10000; 2) studying real-world applications of MMSTSP model is an important research, it is necessary to extend the applications of MMSTSP; 3) strengthening the practical relevance of VNSHA can be studied by incorporating real-world case studies that highlight its effectiveness in dynamic and

uncertain environments, such as transportation, logistics planning and manufacturing scheduling; 4) many optimization methods and machine learning algorithms, such as reinforcement learning, multi-task learning and integrated learning, can be used to improve the proposed algorithms, the improved methods and strategies in Ref.[36, 39] could be studied for optimization.

Acknowledgements

This research is supported or funded by Shandong Provincial Key Laboratory of Fundamental Software, Shandong Provincial Natural Science Foundation of China under Grant No.ZR2023MF092 and No.ZR2024QG191, Basic Science (Natural Science) Research Project of Colleges and Universities in Jiangsu Province (22KJA520004), The Open Project of Anhui Provincial Key Laboratory of Multimodal Cognitive Computation, Anhui University, No.MMC202307, Foundation of State Key Laboratory of Public Big Data (No.PBD2023-14).

References:

[1] J. Li, M.C. Zhou, Q. Sun, et al, Colored traveling salesman problem, IEEE Transactions on Cybernetics. 45(11) (2015) 2390-2401.

[2] L.R. John, The bottleneck traveling salesman problem and some variants, Master of Science of Simon Fraser University, Canada, (2010) 21-23.

[3] W.Y. Dong, X.S. Dong, Y.F. Wang, The improved genetic algorithms for multiple maximum scatter traveling salesperson problems, China Conference on Wireless Sensor Networks (CWSN 2017), CCIS, 812 (2018) 155-164.

[4] Z.H. Ahmed, A hybrid genetic algorithm for the bottleneck traveling salesman problem, ACM Transactions on Embedded Computing Systems. 12(1) (2013) 9:1-9:10.

[5] X.S. Dong, W.Y. Dong, Y.L. Cai, Ant colony optimization for colored traveling salesman problem by multi-task learning, IET Intelligent Transport Systems. 12(8) (2018) 774-782.

[6] X.S. Dong, Y.L. Cai, A novel genetic algorithm for large scale colored balanced traveling salesman problem, Future Generation Computer System. 95 (2019) 727-742.

[7] Q. Cai, M.G. Gong, L.J. Ma, et al, Greedy discrete particle swarm optimization for large-scale social network clustering, Information Sciences. 316 (2015) 503-516.

[8] C. Segura, C.A. Coello, A.G. Hernández-Díaz, Improving the vector generation strategy of differential evolution for large-scale optimization, Information Sciences. 323 (2015) 106-129.

[9] R. Cheng, Y.C. Jin, A competitive swarm optimizer for large scale optimization, IEEE Transaction on Cybernetics. 45(2) (2015) 191-204.

[10] X.S. Dong, Hybrid ITÖ algorithm for large-scale colored traveling salesman problem, Chinese Journal of Electronics, 33(6)(2024)1337-1345.

[11] Q. Sheng, J. Zou, S.X. Yang, J.H. Zheng, A level-based multi-strategy learning swarm optimizer for large-Scale multi-objective optimization, Knowledge-Based Systems. 73(2022) 101100.

[12] M.N. Omidvar, M.Yang, Y. Mei, et al, DG2: a faster and more accurate differential grouping for large-scale black-box optimization, IEEE Transaction on Evolutionary Computation. 21(6) (2017) 929-942.

[13] Q.Yang, W.N. Chen, T.L. Gu, Zhang, et al, Segment-based predominant learning swarm optimizer for large-scale optimization, IEEE Transaction on Cybernetics. 47(9) (2017) 2896-2909.

[14] P. Wang, C.H. Shen, A. Hengel, P.H.S. Torr, Large-scale binary quadratic optimization using semidefinite relaxation and applications, IEEE Transaction on Pattern Analysis and Machine Intelligence. 39(3) (2017) 470-485.

[15] Y. Z. Qiu, L. Wang, X.L. Xu, et al, A variable neighborhood search heuristic algorithm for production routing problems. Applied Soft Computing.66 (2018) 311-318.

[16] S. Hore, A. Chatterjeeb, A. Dewanji, Improving variable neighborhood search to solve the traveling salesman problem, Applied Soft Computing. 68 (2018) 83-91.

[17] X.S. Dong, M. Xu, Q. Lin, et al, ITÖ algorithm with local search for large scale multiple balanced traveling salesmen problem, Knowledge-Based Systems. 229 (2021) 107330.

[18] X.S. Dong, H. Zhang, M. Xu, et al, Hybrid genetic algorithm with variable neighborhood search for multi-scale multiple bottleneck traveling salesmen problem, Future Generation Computer Systems. 114 (2021) 229-242.

[19] X.P. Xu, J. L, M.C. Zhou, Delaunay-triangulation-based variable neighborhood search to solve large-scale general colored traveling salesman problems, IEEE Transactions on Intelligent Transportation Systems. 22(3) (2021) 1583-1593.

[20] J. Pasha, M.A. Dulebenets, A.M. Fathollahi-Fard et al, An integrated optimization method for tactical-level planning in liner shipping with heterogeneous ship fleet and environmental considerations, Advanced Engineering Informatics. 48 (2021) 101299.
[21] Z.Z. Liu, Y. Wang, P.Q. Huang, A many-objective evolutionary algorithm with angle-based selection and shift-based density estimation, Information Sciences. 509 (2020) 400-419.
[22] M.A. Dulebenets, An adaptive polyploid memetic algorithm for scheduling trucks at a cross-docking terminal, Information Sciences. 565 (2021) 390-421.
[23] A.M. Fathollahi-Fard, M.A. Dulebenets, et al, Two hybrid meta-heuristic algorithms for a dual-channel closed-loop supply chain network design problem in the tire industry under uncertainty, Advanced Engineering Informatics. 50 (2021) 101418.
[24] J.F. Rodrigues-Jr, M.A. Gutierrez, G. Spadon, et al, LIG-Doctor: Efficient patient trajectory prediction using bidirectional minimal gated-recurrent networks, Information Sciences. 545(2021) 813-827.
[25] J. Sun, Z. Miao, D. Gong, et al, Interval multiobjective optimization with memetic algorithms, IEEE Transactions on Cybernetics. 50(8) (2020)3444-3457.
[26] Z. Fan, G. Hu, X. Sun, G. Wang, et al, Self-attention neural architecture search for semantic image segmentation, Knowledge-Based Systems. 239(2022) 107968.
[27] Q. Zhai, Y. He, G. Wang, X. Hao, A general approach to solving hardware and software partitioning problem based on evolutionary algorithms, Advances in Engineering Software. 159(2021)102998.
[28] H. Zhao, X. Sun, J. Dong, H. Yu, G. Wang, Multi-instance semantic similarity transferring for knowledge distillation, Knowledge-Based Systems. 256(2022)109832.
[29] S. Chen, R. Chen, G.G. Wang, et al, An adaptive large neighborhood search heuristic for dynamic vehicle routing problems, Computers and Electrical Engineering. 67(2018) 596-607.
[30] F.Q. Zhao, S. Qin, Y. Zhang, et al, A hybrid biogeography-based optimization with variable neighborhood search mechanism for no-wait flow shop scheduling problem, Expert Systems With Applications. 126 (2019) 321-339.
[31] C. Peng, Q. Cheng, Discriminative ridge machine: a classifier for high-dimensional data or imbalanced data, IEEE Transactions on Neural Networks and Learning Systems. 32(6) (2021) 2595-2609.
[32] C. Peng, Y. Chen, Z. Kang, et al, Robust principal component analysis: a factorization-based approach with linear complexity, Information Sciences. 513 (2020) 581-599.
[33] A. Ouali, D. Allouche, S.D. Givry, Variable neighborhood search for graphical model energy minimization, Artificial Intelligence. 278, (2020) 103194.
[34] V. Mnih, K. Kavukcuoglu, D. Silver, et al, Human-level control through deep reinforcement learning, Nature. 518(7540) (2015) 529-33.
[35] H. Peng, S. Zhang, L. Li, et al, Multi-strategy multi-modal multi-objective evolutionary algorithm using macro and micro archive sets, Information Sciences. 663 (2024) 120301.
[36] X. Xu, J. Li, M. Zhou, X. Yu, Precedence-constrained colored traveling salesman problem: an augmented variable neighborhood search approach, IEEE Transactions on Cybernetics. 52(9) (2021) 9797-9808.
[37] H. Peng, C. Mei, S. Zhang, et al, Multi-strategy dynamic multi-objective evolutionary algorithm with hybrid environmental change responses, Swarm and Evolutionary Computation. 82 (2023) 101356.
[38] Y.M. Zhou, W.Q. Xu, M.C. Zhou, Z.H. Fu, Bi-trajectory hybrid search to solve bottleneck-minimized colored traveling salesman problems, IEEE Transactions on Automation Science and Engineering. 21(1)(2024) 895-905.
[39] Z.Z. Zhang, H. Liu, M.C. Zhou, J.H. Wang, Solving dynamic traveling Salesman problems with deep reinforcement learning, IEEE Transactions on Neural Networks and Learning Systems.34(4)(2023) 2119-2132.
[40] Y.M. Zhou, W.Q. Xu, Z.H. Fu, M.C. Zhou, Multi-neighborhood simulated annealing-based iterated local search for colored traveling salesman problems, IEEE Transactions on Intelligent Transportation Systems. 23(9)(2022) 16072-16082.
[41] X.S. Dong, Q. Lin, W. Wang, Hybrid ITÖ algorithm for maximum scatter colored traveling salesman problem, The Computer Journal. 67(6)(2024) 2172-2188.
[42] X.S. Dong, L.W. Ma, X. Zhao, Y.C. Shan, J. Wang, Z.H. Xu, Hybrid genetic algorithm with wiener process for multi-scale colored balanced traveling salesman problem, Expert Systems with Applications. 262(2025) 125610.

Xueshi Dong

Xueshi Dong is currently an associate professor in College of Computer Science and Technology, Qingdao University. He obtained Ph.D degree in School of Computer Science, Wuhan University, and received B.S. and M.S. degree in computer science from Nanjing University of Information Science & Technology. He worked in Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences from 2011 to 2013, and also visited Institute of Computing Technology, Chinese Academy of Sciences and Peking University from 2015 to 2016. His current research interests include intelligent transportation, intelligent computing and simulation optimization.

Fanfan Shen

Fanfan Shen is currently an associate professor in School of Computer Science, Nanjing Audit University. He obtained doctor degree in School of Computer Science, Wuhan University, and received B.S. and M.S. degree from China Three Gorges University and Guilin University of Technology. His research interests include artificial intelligence systems, intelligent auditing embedded system and computer architecture.

Lingling Zhang

Lingling Zhang is research associate in Business School, Shandong University. She obtained Ph.D degree from Dalian University of Technology, and received B.S. and M.S. degree from Harbin Institute of Technology and Dalian University of Technology. Her main research interests include operations research optimization and intelligent decision-making, cloud computing and information systems, supply chain management.

Zhenghao Xu

Zhenghao Xu is pursuing M.S. degree in College of Computer Science and Technology, Qingdao

University. His main research interests include intelligent computing and intelligent transport.

Yongchang Shan

Yongchang Shan studied in College of Computer Science and Technology, Qingdao University. His main research interests include intelligent computing and intelligent transport.

Qing Lin

Qing Lin is an assistant professor in College of Computer Science and Technology, Qingdao University. His main research interests include intelligent computing and intelligent transport.