Intuitionistic Fuzzy Hybrid Discrete Particle Swarm Optimization for Solving Travelling Salesman Problem

Hai-Tao Mei, Ji-Xue Hua, Yi Wang, Tong Wen
(Air and Missile Defense College, Air Force Engineering University, Xi’an 710051, China)
meiht14@163.com, huajx@163.com, Wangy@163.com

Keywords: intuitionistic fuzzy set, membership degree function, identical factor, the local optima, elitist reserving

Abstract. An intuitionistic fuzzy hybrid discrete particle swarm optimization (IF-HDPSO) is proposed for solving travelling salesman problem (TSP). By defining intuitionistic fuzzy charisma function, the IF-HDPSO algorithm exploits some other individuals to participate in the updates of velocity and position except the best one. In addition, the PSO identical factor function is defined to adjust inertia weight and learning operator adaptively, so the IF-HDPSO can explore the whole optimal solution quickly. Furthermore, an adaptive genetic algorithm based on elitist reserving strategy is developed, and combine it with PSO to reduce the probability of being trapped in the local optima and premature convergence. The simulation results indicate IF-HDPSO perform better on precision, iteration number and computational robustness.

Introduction
TSP is a famous combinatorial optimization problem, which is a NP-hard problem and cannot be solved within specific time scale by any known algorithm. Traditional algorithms existed for TSP such as stage dynamic programming, linear programming solution, and greedy method will get into the “combinational explosion problem” with the increase of city number. In recent years, people proposed many intelligence optimization algorithms, such as GA, ACO [1], PSO, firefly algorithm (FA) [2], and cuckoo search algorithm (CSA) [3]. Though these algorithms cannot ensure to find the best solution, they decrease the solution space and increase the probability of explores the optima.

The standard PSO was firstly proposed to solve the continuous space combinatorial optimization problems, as many optimization problems are defined in a discrete space, e.g. TSP and 0-1 knapsack problem. People proposed many improved PSO to solve the discrete problems. N. Salmani Niasar proposed a discrete fuzzy particle swarm optimization [4]. Except the best particle of the swarm; some other particles will be selected to participate in updating according to their degree of charisma.

In this paper, we propose an novel algorithm (IF-HDPSO) for solving TSP. We evaluate each particle’s charisma with the intuitionistic fuzzy fitness, and define identical factor to adjust the parameters of PSO, including inertia weight and learning operator. Furthermore, the hybrid algorithm of an adaptive GA based on elitist reserving strategy and PSO is used to search the optima.

The rest of this paper is organized as follows. Section 2 briefly introduces TSP. Section 3 describes the proposed algorithm. Section 4 presents the result of a set of benchmarks of TSP from the TSPLIB in detail. Finally, Section 5 concludes with some discussions.

Representation of TSP

The representation of TSP is simple, assuming a TSP that is represented as $e = \{c_1, c_2, \ldots, c_n\}$, where $n$ is the number of city, $c_i$ ($i = 1, 2, \ldots, n$) is the $i$-th visited directed edges in turn, the objective is to find a shortest path which visits each city exactly once [5], where $d(c_i, c_j)$ means the distance between $c_i$ and $c_j$ ($c_i, c_j \in e$). So, the cost of a permutation can be defined as follows:
\[
\min f = \sum_{i=1}^{n-1} d(c_i, c_{i+1}) + d(c_n, c_1)
\] 

Intuitionistic Fuzzy Hybrid Discrete PSO

Discrete fuzzy PSO (D-FPSO) [4] which differ from the standard PSO is that the D-FPSO allowed some other particles, not just the best one, to participate in the updating of position and velocity based on their fuzzy charisma. Where the fuzzy charisma \( \psi(h) \) is a fuzzy variable, and represented by the Cauchy membership function as formula (2).

\[
Cauchy(x; \alpha, \beta) = \frac{1}{\beta \sqrt{\pi}} \frac{\alpha}{\alpha^2 + x^2}
\]

Therefore, the position and velocity formula of D-FPSO should be modified as:

\[
v_{id}(t+1) = \omega v_{id}(t) + c_1 r_1 (p_{id} - x_{id}(t)) + \sum_{h \in B(i,k)} c_2 r_2 \text{IF}(h)(p_{gd} - x_{id}(t))
\]

Where \( B(i,k) \) means a set of k best particles in the neighborhood of i-th particle, \( MF(h) \) represents the fuzzy membership function of particle \( h \) in \( B(i,k) \). The D-FPSO can solve the discrete TSP in some point, however, no specific methods can confirm the value of \( k \), which need rich experiences and simulations, and it also will get into the local optima and premature convergence easily.

On the basis of D-FPSO[4], this paper proposes a hybrid discrete PSO (IF-HDPSO), we figure up the intuitionistic fuzzy membership in the neighborhoods of each particle and the intuitionistic fuzzy distance with the best particle of the whole swarm. Then, the distance which is less than the average of particle swarm will be allowed to enter into \( B(i,k) \), and we defined an identical factor function of the swarm to adjust the inertia weight and learning operator adaptively, which can accelerate convergence speed of IF-HDPSO.

A. Intuitionistic fuzzy discrete PSO

Assuming a TSP instance, the objective function’s fitness of particle \( i \) is represented as \( f(x'_i) \), where \( f_{\text{max}}(x'_{id}) \) and \( f_{\text{min}}(x'_{id}) \) mean the max and min fitness of \( i \)-th iteration. We estimate intuitionistic fuzzy charisma of particle \( i \) by the distance with the best one, the intuitionistic membership and nonmembership function of particle \( i \) are described as follows:

\[
\mu(x'_i) = \begin{cases} 
0, & f(x'_i) < f_{\text{min}}(x'_j) \\
\frac{f(x'_i) - f_{\text{min}}(x'_j)}{f_{\text{max}}(x'_j) - f_{\text{min}}(x'_j)}, & f_{\text{min}}(x'_j) < f(x'_i) < f_{\text{max}}(x'_j) \\
1, & f_{\text{max}}(x'_j) < f(x'_i) 
\end{cases} \\
\gamma(x'_i) = \begin{cases} 
0, & f_{\text{max}}(x'_j) - f(x'_i) < f_{\text{min}}(x'_j) - f(x'_i) \\
\frac{f_{\text{max}}(x'_j) - f(x'_i)}{f_{\text{max}}(x'_j) - f_{\text{min}}(x'_j)}, & f_{\text{max}}(x'_j) - f(x'_i) < f(x'_i) < f_{\text{min}}(x'_j) \\
1, & f(x'_i) < f_{\text{min}}(x'_j) 
\end{cases}
\]

The intuitionistic fuzzy charisma of particle \( i \) in \( t \)-th iteration can be described as an intuitionistic fuzzy number, for example \( A = \{<x, \mu(x'_i), \gamma(x'_i)>, \} \), and the best particle of the swarm can be described as \( G = \{<p_g, 1, 0>, \} \) [6], the distance between particle \( i \) and \( p_g \) is \( d_{ig} \).

\[
d_{ig} = \frac{1}{3n} \sum_{i=1}^{n} [(\mu(x'_{ig}) - 1)^2 + (\gamma(x'_{ig}) - 0)^2 + (1 - \mu(x'_{ig}) - \gamma(x'_{ig}))^2]
\]

The average distance of the whole swarm at iteration \( t \) to \( p_g \) is \( d_i \).

\[
d_i = \frac{1}{n} \sum_{i=1}^{n} d_{ig}
\]

\( k \) is a dynamic variable in set of \( B(i,k) \), which means the \( k \) shortest particles set.

\[
d_{ig} \in B(i,k), \quad \text{if } d_{ig} \leq d_i \]
\[
d_{ig} \notin B(i,k), \quad \text{else}
\]

Therefore, the velocity will be modified as below.

\[
v_{id}(t+1) = \omega v_{id}(t) + c_1 r_1 (p_{id} - x_{id}(t)) + \sum_{h \in B(i,k)} c_2 r_2 \text{IF}(h)(p_{gd} - x_{id}(t))
\]
Where $IMF(h)$ is the intuitionistic fuzzy fitness of particle $h$ in set $B(i,k)$, and it can be calculated by Eq. (4).

**B. Self-adaptive PSO algorithm**

In PSO, the inertia weight can affect convergence speed and solution precision greatly [7], in early stage, we should increase the value of $\omega$ to enhance the global search ability. Other hands, in the later period of PSO, we should decrease $\omega$ to enhance the local search ability. $c_1$ and $c_2$ are two so called parameters to weigh the importance of self-cognitive and social-influence, so we should take bigger $c_2$ and smaller $c_1$ in early stage, which can enhance the diversity of the swarm. On the contrary, we should take bigger $c_1$ and smaller $c_2$ to enhance the local search ability and the convergence speed in the later period.

In the early stage of PSO, there are fewer identical individuals and the gap of fitness cannot be ignored. Adversely, identical individuals get increased with the process of iteration, we define identical factor $s_v$ to describe the diversity of particle swarm, which can adjust the inertial weight adaptively.

\[
s_v = 1 + \frac{a}{N} \sum_{i=1}^{N} \left( f_i - f_{avg} \right)
\]

In formula (9), where $N$ is the population scale, Obviously, $s_v \in (0,1)$, $f_i$ and $f_{avg}$ represent the fitness of particle $i$ of $t$-th iteration and average fitness of the whole swarm. $a = 0.1 \times N$ is modulation parameter. Afterwards, we define identical factor function $\theta_s = 1 - \cos\left(\frac{\pi}{2} s_v\right)$

Therefore, the adaptively parameter $\omega$, $c_1$ and $c_2$ can be modified.

\[
\omega^{t+1} = \omega_{max} - (\omega_{max} - \omega_{min}) \cdot \theta_s, \quad c_1^{t+1} = c_1^{max} - (c_1^{max} - c_1^{min}) \cdot \theta_s, \quad c_2^{t+1} = c_2^{max} - (c_2^{max} - c_2^{min}) \cdot \theta_s
\]

Where $\omega \in [0.4, 0.9]$, $c \in [0.5, 2.5]$, we take C-TSP instance as an example, the city path solution searched by standard PSO and self-adaptively PSO is illustrated in Figure 1.

![Fig.1. The comparison between self-adaptive PSO with standard PSO](image)

**C. Hybrid PSO algorithm**

Like other intelligence optimization algorithm, PSO also suffers the disadvantages of premature convergence and get trapped into local optima, thought it can be realized easily and has high convergence speed. To enhance the standard PSO’s ability of jump out the local optima, we propose an adaptive GA based on elitist reserving strategy, and combined it with PSO algorithm to solve the discrete TSP.

Procedures of GA mainly include selection, crossover and mutation, which are important methods to keep elitist and engender new individuals. We bring GA procedures into PSO, reserved the elitist individuals of the swarm and take $p_{el_p}, p_{el_d}$ as the parent individuals, then, we employ the adaptive selection, crossover and mutation operator to update $p_{el_p}, p_{el_d}$ by comparing the fitness of GA procedures around. This strategy can duplicate the excellent individuals and reject the worst particle.

Selection: Firstly, we reserve the best-so-far solution obtained by the whole swarm, and stores it in the set of crossover. Then, selecting other parent individuals by roulette wheel strategy, the
selected probability is \( p_{bh} \).

\[
p_{bh} = \frac{f(x'_b)}{\sum_{i=1}^{n} f(x'_i)}
\]  

(11)

Crossover: \( p_c \) can affect the performance of GA directly. If crossover probability is very large, the excellent gene of parent individuals will be destroyed, and it will lead to convergent slowly. We perform crossover with adaptive probability.

\[
p_c = p_{cmin} + (p_{cmax} - p_{cmin}) \frac{f(x'_i) - f_{min}(x'_d)}{f_{max}(x'_d) - f_{min}(x'_d)} = p_{cmin} + (p_{cmax} - p_{cmin}) \mu(x'_i)
\]

(12)

Where \( p_{cmin} \) and \( p_{cmax} \) are the minimum and maximum crossover probability.

Mutation: Because the result of mutation operator is unknown, an appropriate mutation probability can increase the diversity of swarm and avoid being trapped in the local optima. We perform mutation with adaptive probability.

\[
p_m = p_{mmin} + (p_{mmax} - p_{mmin}) \mu(x'_i)
\]

(13)

Elitist reserving strategy: Elitist reserving strategy can store \( p_{gd} \) and take it as the parent individuals directly. Then, replacing the worst individual obtained by GA with \( p_{gd} \) after the crossover and mutation operation.

**Experimental Results**

The proposed IF-HDPSO algorithm is tested by some instances of TSP taken from the publicly available electronic library TSPLIB of TSP problems [8]. Most of the instances in the TSPLIB had already been solved by other algorithms and their optimal can be used to compare the performance of novelty algorithms. The comparison among GA, PSO, ACO, SA and IF-HDPSO are carried out.

We have set the parameter of proposed algorithm as, the population scale is 200 and iteration number is 300. Where basic parameter \( \omega \in [0.4, 0.9] \) and \( c \in [0.5, 2.5] \). The operator of GA, we take \( p_c \in [0.4, 0.9] \) and \( p_m \in [0.01, 0.1] \). For the justice and corrective of the experiment, we set the same test environment of GA, PSO, ACO, SA with IF-HDPSO, some other parameters are listed in Table 1.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameter settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA</td>
<td>( p_c = 0.8, \ p_m = 0.02 )</td>
</tr>
<tr>
<td>PSO</td>
<td>( \omega = 0.75, \ c_i = c_t = 1.95, \ n \in [-0.5, 0.5] )</td>
</tr>
<tr>
<td>ACO</td>
<td>( \alpha = 3, \ \beta = 2, \ \rho = 0.3, \ Q = 5 )</td>
</tr>
<tr>
<td>SA</td>
<td>( T_c = 1000, \ T_{end} = 0.001, \ q = 0.9, \ L = 200 )</td>
</tr>
</tbody>
</table>

In order to test the performance of IF-HDPSO algorithm with different city scale, we download 10 TSP instances from TSPLIB and test each problem for 20 runs. Table 2 lists the instance name, optimal solution in TSPLIB and the comparison result of 5 algorithms. The figure in the name of an instance represents the number of provided cities. For example, att48 provides 48 cities with their coordinates. Figure 3 shows the path planning of att48, eil51, eil101, ch150 and a280 searched by IF-HDPSO.
<table>
<thead>
<tr>
<th>Instance</th>
<th>Optimal in TSPLIN</th>
<th>IF-HDPSO</th>
<th>GA</th>
<th>PSO</th>
<th>ACO</th>
<th>SA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iteration number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bays29</td>
<td>9 074</td>
<td>18</td>
<td>37</td>
<td>29</td>
<td>40</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>Result</td>
<td>9 074</td>
<td>9 074</td>
<td>9 074</td>
<td>9 178</td>
<td>9 074</td>
</tr>
<tr>
<td></td>
<td>Deviation rate</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.14</td>
<td>0</td>
</tr>
<tr>
<td>att48</td>
<td>33 522</td>
<td>25</td>
<td>54</td>
<td>41</td>
<td>37</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>Result</td>
<td>33 522</td>
<td>33 966</td>
<td>33 784</td>
<td>33 522</td>
<td>34 623</td>
</tr>
<tr>
<td></td>
<td>Deviation rate</td>
<td>0</td>
<td>1.32</td>
<td>0.78</td>
<td>0</td>
<td>3.28</td>
</tr>
<tr>
<td>eil51</td>
<td>426</td>
<td>31</td>
<td>51</td>
<td>49</td>
<td>62</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>Result</td>
<td>426</td>
<td>437</td>
<td>429</td>
<td>442</td>
<td>426</td>
</tr>
<tr>
<td></td>
<td>Deviation rate</td>
<td>0</td>
<td>2.58</td>
<td>0.70</td>
<td>3.75</td>
<td>0</td>
</tr>
<tr>
<td>pr76</td>
<td>108 159</td>
<td>39</td>
<td>79</td>
<td>68</td>
<td>80</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td>Result</td>
<td>108 159</td>
<td>114</td>
<td>110</td>
<td>108 354</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td>Deviation rate</td>
<td>0</td>
<td>5.90</td>
<td>1.83</td>
<td>0.18</td>
<td>1.88</td>
</tr>
<tr>
<td>kro100</td>
<td>22 141</td>
<td>46</td>
<td>95</td>
<td>86</td>
<td>88</td>
<td>101</td>
</tr>
<tr>
<td></td>
<td>Result</td>
<td>22 141</td>
<td>22 508</td>
<td>22 369</td>
<td>23 584</td>
<td>22 196</td>
</tr>
<tr>
<td></td>
<td>Deviation rate</td>
<td>0</td>
<td>1.65</td>
<td>1.02</td>
<td>6.51</td>
<td>0.25</td>
</tr>
<tr>
<td>gr120</td>
<td>6 942</td>
<td>61</td>
<td>106</td>
<td>96</td>
<td>115</td>
<td>121</td>
</tr>
<tr>
<td></td>
<td>Result</td>
<td>6 942</td>
<td>7 083</td>
<td>7 009</td>
<td>6 984</td>
<td>7 049</td>
</tr>
<tr>
<td></td>
<td>Deviation rate</td>
<td>0</td>
<td>2.03</td>
<td>0.96</td>
<td>0.60</td>
<td>1.54</td>
</tr>
<tr>
<td>ch130</td>
<td>6 111</td>
<td>77</td>
<td>116</td>
<td>103</td>
<td>137</td>
<td>142</td>
</tr>
<tr>
<td></td>
<td>Result</td>
<td>6 143</td>
<td>6 506</td>
<td>6 127</td>
<td>6 148</td>
<td>6 129</td>
</tr>
<tr>
<td></td>
<td>Deviation rate</td>
<td>0.52</td>
<td>6.46</td>
<td>0.26</td>
<td>0.61</td>
<td>0.29</td>
</tr>
<tr>
<td>ch150</td>
<td>6 528</td>
<td>92</td>
<td>122</td>
<td>125</td>
<td>159</td>
<td>154</td>
</tr>
<tr>
<td></td>
<td>Result</td>
<td>6 528</td>
<td>6 842</td>
<td>6 867</td>
<td>6 665</td>
<td>6 547</td>
</tr>
<tr>
<td></td>
<td>Deviation rate</td>
<td>0</td>
<td>4.81</td>
<td>5.19</td>
<td>2.01</td>
<td>0.29</td>
</tr>
<tr>
<td>pr226</td>
<td>80 369</td>
<td>133</td>
<td>168</td>
<td>179</td>
<td>183</td>
<td>169</td>
</tr>
<tr>
<td></td>
<td>Result</td>
<td>80 369</td>
<td>85 254</td>
<td>82 913</td>
<td>83 983</td>
<td>81 768</td>
</tr>
<tr>
<td></td>
<td>Deviation rate</td>
<td>0</td>
<td>6.01</td>
<td>3.17</td>
<td>4.50</td>
<td>1.74</td>
</tr>
<tr>
<td>a280</td>
<td>2 579</td>
<td>157</td>
<td>172</td>
<td>192</td>
<td>188</td>
<td>196</td>
</tr>
<tr>
<td></td>
<td>Result</td>
<td>2 579</td>
<td>2 815</td>
<td>2 668</td>
<td>2 662</td>
<td>2 606</td>
</tr>
<tr>
<td></td>
<td>Deviation rate</td>
<td>0.19</td>
<td>9.15</td>
<td>3.45</td>
<td>3.22</td>
<td>1.05</td>
</tr>
</tbody>
</table>
In Table 2, the proposed IF-HDPSO performs better in the iteration number and optimal solution compared with other 4 algorithms. Especially, for the bays29, 5 algorithms all can search the global optimal, the superiority of IF-HDPSO is not remarkable, but it needs fewer iteration number obviously. After analysis, the reason indicates IF-HDPSO algorithm adopts identical factor function to adjust inertia weight and learning factor, moreover, the GA based on elitist reserving strategy hybrid PSO can avoid IF-HDPSO being trapped in local optima and premature convergence, and guide the fly experience of particle to the global optima. Except ch130, it can get the known best solution of TSPLIB, and stabilize at the solution for many times. The path planning searched by other algorithms have many crosses, which means these algorithm is trapped into the local optima and stopped searching. Generally, for the complicated and discrete TSP instances, ACO and SA possess quicker convergence speed compared with GA and PSO, and they can seek out the better path planning, however, when the dimension is greater than 100, due to the complication of TSP and limitation of the algorithm themselves, they always cannot get better result, such as pr226 and a280 instance. On the other hand, IF-HDPSO algorithm can get the best solution within specified iteration number, which demonstrates IF-HDPSO also has great capability in solving large scale TSP instances.

As the path planning shown in Figure 2, the two cities with shorter distance have greater probability to joint together, the patterning is always raised polygon and not exists intersected node. The proposed IF-HDPSO algorithm, particles get closed to the global optima by the learning factor \( c \) continuously, the identical factor function can adjust \( c \) so that avoid being trapped in local optima, thus, in the next city selection, it has greater probability to select near city. Intuitionistic fuzzy charisma function allows other particles to participate in the update of velocity and position except the best one, which also can increase the diversity of particle swarm. The elitist reserving strategy preserves admirable individuals in case being destroyed, and replace the low fitness individuals, which can decrease the useless search procedure and the proposed IF-HDPSO needs fewer iterations.

Summary

In this paper, we propose a hybrid method IF-HDPSO, and utilize it to discrete space problem TSP. We apply intuitionistic fuzzy charisma function to represent the corresponding TSP solution, which broadens the application in discrete space. The intuitionistic fuzzy charisma function can select other particles to participate in velocity and position updating. Identical factor function is
defined to measure the diversity of swarm and adjust the inertia weight and learning factor. The hybrid of elitist reserving GA and PSO enhance the global search ability and local escape ability. The experimental results indicate IF-HDPSO algorithm is more confidential than other 4 algorithms, and it also can be utilized in discrete spaces.

Acknowledgments

The authors would like to thank the anonymous referees for their careful reading of their helpful comments. This work was supported in part by the National Science Foundation of China under project 61402517, in part by the National Science Foundation of shanxi province under project 2013JQ8035. Hai-tao Mei is the corresponding author. (email: meiht14@163.com)

References