Grouping memetic search for the colored traveling salesmen problem

Pengfei HE [1)], Jin-Kao HAO [1)], Qinghua WU [2)]

1) LERIA, Université d'Angers, 2 Boulevard Lavoisier, 49045 Angers Cedex 01, France
2) School of Management, Huazhong University of Science and Technology, Wuhan, China

Mots-clés : *Memetic algorithm, local search, colored traveling salesman problem.*

1 Introduction

The colored traveling salesmen problem (CTSP) can be stated as follows [5]. Given a undirected graph G=(V, A) with vertex set $V = \{0, 1, 2, \cdots, n\}$ where 0 is the depot while other vertices $N = \{1, 2, \cdots, n\}$ represent cities and $A = \{\{i, j\} : i, j \in V, i \neq j\}$ is the set of edges. Each edge $\{i, j\} \in A$ has a non-negative weight c_{ij} representing the traveling distance between vertices i and j. The vertices of V are divided into $m + 1$ disjoint sets : m exclusive city sets $\{C_1, C_2, \cdots, C_m\}$, and one shared vertex set S such that $\cup_{k=1}^{m} C_k \cup S = V$ and $\cap_{k=1}^{m} C_i \cap S = \emptyset$. The cities of an exclusive set C_k $(k = 1, 2, \cdots, m)$ are to be visited by salesman k only, while the shared vertices can be visited by any of the m salesmen. Besides, vertex 0 (the depot) belongs to the shared vertex set S and is visited by all salesmen. CTSP is to determine m shortest Hamiltonian tours (routes) starting from the depot and ending at the depot such that each exclusive city in C_k is visited exactly once by salesman k and each shared vertex is visited exactly once by one of the m salesmen.

CTSP generalizes a variant of the classical traveling salesman problem (TSP), known as the multiple traveling salesmen problem (mTSP). As mentioned in [5], CTSP is a useful model for a number of practical problems, such as multi-bridge machining systems and rice harvesters problem. This work investigates CTSP by proposing an effective heuristic.

2 Grouping memetic search for CTSP

We propose a specialized grouping memetic algorithm (GMA) for CTSP [4], which is composed of four main components : population initialization, local optima exploration, backbone-based crossover and population updating. GMA starts with an initial population generated by the population initialization procedure. It then repeats a number of generations during which new candidate solutions are sampled. At each generation, the backbone-based crossover combines two randomly and uniformly selected parents to generate a promising offspring individual, where useful building blocks are transformed from the parents to the offspring. Then, local optimization is applied to explore local optima around the offspring. For an effective examination of local optima, GMA employs a specific strategy that combines an inter-route optimization with the constrained cross-exchange operator and an intra-route optimization with the TSP heuristic called Edge Assembly Crossover [6]. Finally, a surviving strategy is adopted to update the population. GMA stops when an allotted cutoff time limit is reached.

3 Computational results

GMA is assessed based on three sets of of 65 instances with 23–7397 vertices, introduced in [5, 7, 1, 2]. To ensure a fair comparison, we faithfully re-implemented the best ABC algorithm

[7] (denoted by re-ABC). We accomplished an experimental comparison between GMA and the best performing algorithm ITPLS [3] and re-ABC with the same stopping condition. Each algorithm was run 20 times independently to solve each instance with distinct random seeds.

Table 1 summarizes the results reported by the compared algorithms on the three sets of 65 instances. Column 2 gives the set name and the number of instances in the set. Column 3 shows the quality indicators (f_{best} and f_{avg}). Columns 4-6 count the number of instances for which GMA achieves a better, equal or worse value compared with each reference algorithm. The last column presents the *p-values* from the Wilcoxon signed-rank test. Table 1 reveals large performance gaps between GMA and each reference algorithm on Sets II and III. We conclude that GMA is very competitive for solving CTSP and this is particularly true for large instances.

TAB. 1 – Summary of comparative results between GMA and two reference algorithms

Algorithm pair	Set/Instance	Indicator	Better	Equal	Worse	$p - value$
GMA vs. ITPLS	I/20	f_{best}	0	20	0	0.00E+00
		f_{avg}	0	20	0	0.00E+00
	II/14	f_{best}	7	4	3	2.44E-04
		f_{avg}	8	1	5	4.80E-02
	III/31	f_{best}	31	0	0	1.17E-06
		f_{avg}	31	0	0	1.17E-06
GMA vs. re-ABC	I/20	f_{best}	0	20	0	0.00E+00
		f_{avg}	0	20	0	0.00E+00
	II/14	f_{best}	8	4	2	1.37E-02
		f_{avg}	9	1	4	4.79E-02
	III/31	f_{best}	31	0	0	1.17E-06
		f_{avg}	31	0	0	1.17E-06

Références

[1] Xueshi Dong, Wenyong Dong, and Yongle Cai. Ant colony optimisation for coloured travelling salesman problem by multi-task learning. *IET Intelligent Transport Systems*, 12(8) :774–782, 2018.

[2] Xueshi Dong, Qing Lin, Min Xu, and Yongle Cai. Artificial bee colony algorithm with generating neighbourhood solution for large scale coloured traveling salesman problem. *IET Intelligent Transport Systems*, 13(10) :1483–1491, 2019.

[3] Pengfei He and Jin-Kao Hao. Iterated two-phase local search for the colored traveling salesmen problem. *Engineering Applications of Artificial Intelligence*, 97 :104018, 2021.

[4] Pengfei He, Jin-Kao Hao, and Qinghua Wu. Grouping memetic search for the colored traveling salesmen problem. *Information Sciences*, 570 :689–707, 2021.

[5] Jun Li, MengChu Zhou, Qirui Sun, Xianzhong Dai, and Xiaolong Yu. Colored traveling salesman problem. *IEEE Transactions on Cybernetics*, 45(11) :2390–2401, 2014.

[6] Yuichi Nagata and Shigenobu Kobayashi. A powerful genetic algorithm using edge assembly crossover for the traveling salesman problem. *INFORMS Journal on Computing*, 25(2) :346–363, 2013.

[7] Venkatesh Pandiri and Alok Singh. A swarm intelligence approach for the colored traveling salesman problem. *Applied Intelligence*, 48(11) :4412–4428, 2018.