Multi-objective multi-factorial memetic algorithm based on bone route and large neighborhood local search for VRPTW

Zifeng Zhou, Xiaoliang Ma, Zhengping Liang, and Zexuan Zhu*
College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
zhuzx@szu.edu.cn

Abstract—**Multi-tasking optimization (MTO) has attracted increasing attention in the domain of evolutionary computation. Different from single-tasking optimization, MTO can solve multiple optimization tasks simultaneously to improve the performance of solving each optimization task by inter-task knowledge transfer. Multifactorial evolutionary algorithm (MFEA) is one of the most widely used MTO algorithm based on assortative mating and vertical cultural transmission. This work extends MFEA by integrating bone route and large neighborhood local search to solve multi-objective vehicle routing problem with time window (VRPTW). The VRPTW is modeled as two related tasks, i.e., one is a multi-objective version of VRPTW (the main task), and the other is a single-objective version of VRPTW (the auxiliary task). The resultant new algorithm namely multi-objective multi-factorial memetic algorithm (MOMFMA) solve the two tasks simultaneously where the information between the tasks is exchanged in the evolutionary process. In addition to the implicit information transfer of MFEA, the bone route is introduced to enable explicit information transfer between tasks. Particularly, bone routes are constructed as semi-finished product solutions and used in large neighborhood local search. The bone route and the large neighborhood local search work together to speed up the convergence of the algorithm. MOMFMA is tested on Solomon's 56 datasets and the experimental results demonstrate that the efficiency of MOMFMA.**

Index Terms—**multi-tasking optimization; multifactorial evolutionary algorithm; multi-objective optimization; vehicle routing problem**

I. INTRODUCTION

Evolutionary algorithms (EAs) originate from the Darwinian's principle of "survival of the fittest" [1]. They begin with a population of individuals and use some genetic operators, e.g., selection and reproduction, to produce new offspring. Thanks to the simplicity and powerful global search ability, EAs have been applied to solve many complex optimization problems, such as intelligent scheduling [2], path planning [3], and UAV optimization [4]. Most of the existing EAs are single-tasking methods, i.e., the algorithms solve a problem in a single run and they should be run from scratch to solve a new problem. However, in practice, the knowledge or information learned in the course of solving one problem can facilitate the solving of another related problem. Gupta et al. [5] proposed an evolutionary multi-tasking algorithm (EMT) namely multifactorial evolutionary algorithm (MFEA) to deal with multiple optimization tasks simultaneously by transferring knowledge learned from one task to another. EMT algorithms can obtain better overall performance on the related optimization tasks than the counterpart single-tasking algorithms thanks to the potential synergistic effect between tasks.

In this paper, we extend MFEA to solve multi-objective vehicle routing problem with time window (VRPTW). VRPTW is a fundamental path planning problem in logistic industry and it mathematically falls within combinatorial optimization. The goal to solve a VRPTW is to find optimal routes in terms of some specific criterion for multiple vehicles to serve the custom nodes with each node served once by one vehicle. The complexity of VRPTW posts big challenges to the traditional mathematical programming methods. EAs including MFEA have been successfully utilized to solve VRPTW [6-8]. In this work, we solve VRPTW from a new perspective by modeling it into a multi-tasking optimization (MTO) problem, i.e., a multi-objective version of VRPTW and a single-objective version of VRPTW. The formulation also falls within the multi-form optimization [21] in transfer optimization. To solve the problem, we integrate bone route and large neighborhood local search into MFEA resulting in a multi-objective multi-factorial memetic algorithm (MOMFMA). Both tasks are optimized simultaneously by MOMFMA via inter-task information exchange. Bone route [9] is integrated into large neighborhood local search to speed up the convergence of MOMFMA. The performance of MOMFMA is verified on Solomon's 56 datasets and the experimental results show that MOMFMA is comparable or superior to two compared algorithms.

The rest of the article is organized as follows. Section II introduces the concepts of MTO and multi-objective memetic algorithm. Section III presents the formulation of VRPTW. Section IV details the proposed MOMFMA. Section V describes the experimental results. Finally, Section VI concludes this study.

II. PRELIMINARIES

A. Multi-tasking Optimization (MTO)

MTO [6,10-13,22-28] solves K different optimization tasks concurrently to improve the overall performance of the tasks via inter-task knowledge transfer. Suppose the objective function of the $j th$ task T_j is defined as $\boldsymbol{F}_j : X_j \rightarrow R$, where X_j is the search space of T_j. MTO takes the advantage of the

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

parallelism of population-based search to identify the optimal solution:

$$\{\boldsymbol{x}_1^*,\boldsymbol{x}_2^*,\ldots,\boldsymbol{x}_K^*\}=\arg\min\{\boldsymbol{F}_1(\boldsymbol{x}_1),\boldsymbol{F}_2(\boldsymbol{x}_2),\ldots,\boldsymbol{F}_K(\boldsymbol{x}_K)\} \quad (1)$$

where $\boldsymbol{x}_j$ is a feasible solution in X_j. Every optimization problem $\boldsymbol{F}_j$ is considered as a particular task affecting the evolutionary process.

It is essential to encode and decode the individuals suitably to implement transfer knowledge among different tasks. One way to achieve this goal is to encode all individuals in a unified search space Y, and decode them into task-specific solutions. In the multi-tasking environment, a new scheme for individual fitness evaluation is required. Particularly, the following attributes are introduced:

- *Factorial Rank:* On a given task T_j, the factorial rank r_{ij} of an individual p_i is the rank of p_i in the population (in ascending order) in terms of some specific measure.
- *Skill Factor:* The skill factor τ_i of individual p_i is the task on which p_i performs the best, i.e., $\tau_i = \min_{1\le j\le K}\{r_{ij}\}$.
- *Scalar Fitness:* The scalar fitness is $1/r_{ij}$ for the individual p_i on task T_j.

B. Multi-objective Memetic Algorithm

Memetic algorithms (MAs) are widely known as population-based meta-heuristic algorithms or hybrid global-local search, inspired by Darwinian principles of natural selection and Dawkins' notion of a meme defined as a unit of cultural evolution that is capable of local/individual refinements [20]. MAs take advantage of both population-based global search and local search/domain knowledge to achieve better performance than that of using purely global/local search. Multi-objective MA (MOMA) is developed toward multi-objective optimization (MOO) problems in a MA framework as shown in Algorithm 1. The difference between MOMA and traditional EAs lies in the introduction of local search to improve the quality of solution and to accelerate the convergence of the algorithm.

Algorithm 1. The procedure of MOMA

1. Initialize the population
2. **While** stopping conditions are not satisfied **do**:
3. Evaluate every individual in the population
4. Use genetic operators on the population to get an offspring population
5. Apply local search to improve the quality of the individuals
6. Select elite individuals into the next generation
7. **End While**

III. Problem Formulation

This section presents the formulation of vehicle routing problem with time windows (VRPTW). Generally, a VRPTW is defined as follows. Given n customers to be served by a fleet of vehicles, each of which requires a quantity of goods. Every vehicle should depart from the depot, go through several customers, and return to the same depot. Moreover, each vehicle is supposed to have the same capacity, which must be greater or equal to the total of all demands. Moreover, every customer must be served once and only once by one vehicle. The time window is defined in the interval of the earliest arrival time and the latest arrival time, in which each customer is served. Each vehicle needs to complete its route within the time window of the depot. A solution is a collection of vehicle routes. We describe the depot as node 0 and the solution is denoted as: $R=\{0,c_1,c_2,\ldots,c_{n_1-1},c_{n_1},0,c_{n_1+1},c_{n_1+2},\ldots,0,c_{(n_1+\cdots+n_{m-1}+1)},\ldots,c_{(n_1+\cdots+n_m)},0\}$ where $c_i\in\{1,2,\ldots,L\}$ is a customer node. The number of routes m represents the number of vehicles used and can be decided by the number of '0' in the solution. The objectives of VRPTW in this paper are to minimize distance traveled by the vehicles and the total number of vehicles used to serve the customers.

The key idea in this paper is to take the advantage of MTO to solve VRPTW, which is modeled as a two-tasking problem, i.e., a MOO version of VRPTW (main task) and a single-objective version of VRPTW (auxiliary task). Both tasks are optimized simultaneously by the proposed algorithm via inter-task information exchange. The propose of introducing the auxiliary task is to accelerate the main task. Although the auxiliary task might introduce some illegal solutions in the process of search, it provides inspiration for the creation of bone routes and semi-finished product solutions. The algorithm takes advantage of these illegal solutions to improve the search ability.

The two-tasking VRPTW is mathematically defined as follows:

- **Definition of hard time window (main task)**

$$\begin{cases} \min F_1(x)=(f_1,f_2) \\ f_1=m \\ f_2=\sum_{v=1}^{m}\left[\sum_{i=1}^{n_v} d\left(c_{n_v+i},c_{n_v+i+1}\right)\right] \end{cases} \quad (3)$$

- **Definition of soft time window (auxiliary task)**

$$\min F_2(x)=K_1\cdot f_1+K_2\cdot f_2+K_3\cdot P(r_i) \quad (4)$$

subject to

$$\sum_{i=0}^{n_1+\cdots+n_m} w_i y_{is}\le W, s=1,2,3,\ldots,m \quad (5)$$

$$ts_i\le \mathrm{tr}_{is}\le \mathrm{te}_i, s=1,2,\ldots,m, i=1,2,\ldots,l \quad (6)$$

$$P(r_i)=\begin{cases} a*\max\left((ts_i-r_i),0\right) & ,r_i\le ts_i \\ 0 & ,ts_i\le r_i\le te_i \\ b*\max\left((r_i-te_i),0\right) & ,te_i\le S_i \end{cases} \quad (7)$$

The Eqs. (3) and (4) define a MOO version of VRPTW (main task) and a single-objective version of VRPTW (auxiliary task), respectively. Eq. (3) defines the VRPTW with hard time window and includes two objectives, i.e., the number of vehicles named f_1 and distance traveled by vehicles f_2. $d(\square,\square)$ calculates the distance between two customers. Eq. (4) formulates the VRPTW with soft time window, which is focused on minimizing the total cost of all the routes. K_1 and K_2 are the cost coefficients of f_1 and f_2, respectively. K_3 is the penalty factor of the cost of time

Algorithm 2. Procedure of MOMFMA

Input: ***n***: iteration of searching common information; ***N***: the population size;.
Output: The best solution of main task.
1. Randomly generate ***N*** individuals in the unified search space **Y** as the initial population **current-pop**
2. Evaluate each individual $\boldsymbol{p_i}$ in **current-pop** on two optimization tasks (3)-(4)
3. Evaluate the factorial rank of each individual based on non-dominated sorting and crowding distance of NSGAII [14] for task (3) and fitness for task (4)
4. Assign skill factor for each individual
5. **While** stopping criterions are not met **do**
6. Apply genetic operator (Algorithm 3) on **current-pop** to generate an **offspring-pop**
7. Evaluate each individual in **offspring-pop** and combine **current-pop** with **offspring-pop** as **whole-pop**
8. Calculate the factorial rank of each individual in **whole-pop** based on non-dominated sorting and crowding distance of NSGAII [14] for task (3) and fitness for task (4)
9. Apply large neighborhood local search with bone route (Algorithm 4) with sharing pool ***SP*** to each individual of **whole-pop**
10. **For** ***t*** = 1 to ***n*** do
11. Randomly select two individual to share their common information called **bone route *BR*** and put ***it*** into sharing pool ***SP***
    End for
12. Use ***BR*** in ***SP*** to create a semi-finished solution (Algorithm 5) and put it into template pool ***TP***
13. Use stratified sampling selection based on skill factor (Algorithm 6) to form next generation
14. **End while**

constraint violation $P(r_i)$. In this study, the sum of K_1 and K_2 is set to 1. The specific values of the two coefficients depend on the preference of the objectives. If the number of vehicles is considered more important, then K_1 takes larger value than K_2. Otherwise, K_1 takes smaller value than K_2. The range of K_3 is set to [0,1]. The specific value of K_3 depends on the tolerance to the constraint violation. The constraints (5) and (6) define the capacity of vehicle and time window of customer. w_i is the package's weight of customer i and y_{is} indicates the customer is served by vehicle s. W is the weight of vehicle's limitation. te_i and ts_i are the time windows of customer *i*, respectively. tr_{is} is the arrival time of vehicle s. Eq. (7) indicates the cost of violating time constraint, where a and b are the cost coefficients whose values are depended on the penalty for the early and late arrival of vehicle. The ranges of a and b are set to [0,1], respectively. The specific values of a and b represent the punishment strength of time violations.

IV. THE PROPOSED METHOD

In this section, we present the details of the proposed MOMFMA. The pseudocode of MOMFMA is outlined in Algorithm 2. Lines 1-4 are the initialization of the algorithm. In line 1, MOMFMA starts with N randomly initialized individuals. Each individual in the initialized population is evaluated on two tasks in line 2, and the skill factor of each individual is obtained based on non-dominated sorting and crowding distance of NSGAII [14] for the main task (3) and the fitness for the auxiliary task (4). The evolution process is presented from line 5 to line 14. Firstly, the genetic operator defined in Algorithm 3 is used to produce the offspring. Secondly, the parent population and offspring population are combined and each individual's factorial rank is recalculated. Afterward, bone route based large neighborhood local search is applied to each individual in the whole population. Next, in lines 10-11, the bone routes are created as illustrated in Fig 1, where a bone route is a short segment shared by two routes. Line 12 repairs the bone route and turns it into a semi-finished solution that is used in local search. Finally, stratified sampling selection is used to choose the next generation.

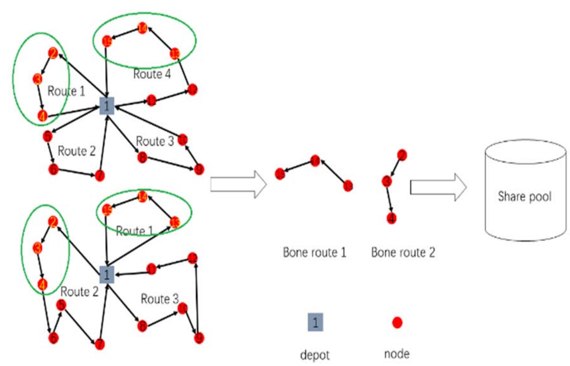

Fig 1. The creation of bone route

An outline of genetic operation is shown in Algorithm 3. In the process of genetic operation, we follow the framework of MFEA's assortative mating [5], in which an individual prefers to mate with another individual with the same skill factor. Contrarily, individuals with different skill factors can only crossover with a prescribed random mating probability.

The bone route based large neighborhood local search is outlined in Algorithms 4. The main idea of the local search on an individual solution is to destroy the solution with one of the following four destroying methods and then repair the solution with one of the two repair methods defined as follows. The repaired solution replaces the original solution, if improvement is achieved.

A. Destroy methods

- **Remove the shortest route (RSR)**: this method removes the shortest route in the solution
- **Remove separated points (RSP)**: this method removes the points with the largest distance to their adjacent points in each route

- **Remove random points (RRP)**: this method randomly removes the points in the individual solution
- **Remove not semi-finished product solution points (*RNSFP*)**: this method deletes the points that are not in the semi-finished solution, and then the semi-finished solution is used to repair by reinsert method

B. Repair methods

- **Reinsert**: to repair the destroyed solution by inserting the removed points with the lowest cost.
- **Exchange**: to repair the destroyed solution by exchanging the positions of the customer with others that are not removed by the destroy methods. Note that this method is not used in the first and the last destroy methods.

Algorithm 3. The genetic operation

Input: population ***P*,** offspring size ***S*,** random mating probability ***rmp***
Output: offspring $\boldsymbol{P_s}$
1. **For *i* = 1** to ***S*** do
2. Select two individuals $\boldsymbol{d_i}$ and $\boldsymbol{d_j}$ from population P
3. **If** $\tau_{d_i} = \tau_{d_j}$ or rand(0,1) < ***rmp***
4. Order-based crossover
5. **Else**
6. **If** rand(0,1) < 0.5
7. Randomly swap two nodes in $\boldsymbol{d_i}$ or $\boldsymbol{d_j}$
8. **Else**
9. Randomly choose a segment **U** in $\boldsymbol{d_i}$ or $\boldsymbol{d_j}$ and reverse it
10. **End For**
11. **Return** $\boldsymbol{P_s}$

Algorithm 4. Bone route based large neighborhood local search

Input: population ***P*** ; population size ***S*** ; template pool ***TP***
Output: an improved population ***IP***
1. **For *i* = 1** to ***S* do**
2. Randomly select an destroy method from four methods, i.e., remove the shortest route (***RSR***), remove separated points (***RSP***), remove random points (***RRP***), and remove not semi-finished product solution points (***RNSFP***) to break individual $\boldsymbol{d_i}$ into $\boldsymbol{d_i'}$
3. Randomly select a repair method from reinsert and exchange to repaired ***di'*** into $\boldsymbol{d_i''}$
4. Evaluate $\boldsymbol{d_i''}$ according to its skill factor
5. **If** $\boldsymbol{d_i''}$ is better than $\boldsymbol{d_i}$ **then**
6. $\boldsymbol{d_i = d_i''}$
7. **If** $\boldsymbol{d_i''}$ is better than the global optimal ***d*** **then** $\boldsymbol{d = d_i''}$
8. **End For**
9. **Return *IP***

The process of creating semi-finished product solutions is shown in Algorithm 5. Every time the algorithm selects the lowest cost bone route from the sharing pool and then extends it to a complete route. We only repair the bone route at the head and the tail in order to ensure the originality of the sharing information. Broadcasting the bone route to the whole population is to use the population diversity and reduce time to repair the bone route. The procedure of creating semi-finished product solution is also depicted in Fig 2.

Algorithm 5. Create a semi-finished product solutions

Input: population ***P*** ; shared pool ***SP***
Output: semi-finished product solutions ***SPS***
1. Sort the bone routes in ***SP*** in descending order
2. **While** exits unused bone routes **do**
3. Select the first unused bone route ***BR***
4. **While** exits unused bone routes **do**
5. Find the previous node p_n or next node *nn* that meets constraint of head node *h* or tail node *t* of ***BR*** from ***P***, and put the found node into set ***H*** or set ***T***
6. Sort set ***H*** (set ***T***) in descending order by distance between p_n (*nn*) and *h* (*t*)
7. Select the most suitable node to insert in front of *h* (or behind *t*)
8. **End While**
9. Get the complete path ***BRP*** of one vehicle through the above operations
10. **End While**
11. Connect all the ***BRP***s to get a semi-finished product solutions
12. **Return *SPS***

Algorithm 6. Stratified sampling selection based on skill factor

Input: population ***P*** ; offspring size ***S***
Output: next generation ***GP***
1. Classify the individuals in ***P*** into two sets ***A*** and ***B*** according to skill factor
2. Sort set A and set B in descending order by fitness respectively
3. Divide set ***A*** (set ***B***) into three equal parts
4. According to the relationship of 3:2:1, the offspring is extracted from three equal parts to form ***GP***
5. **Return *GP***

To maintain the diversity of population, we use stratified sampling selection based on skill factor as shown in Algorithm 6. The purpose of this selection method is to increase the possibilities of creating semi-finished product solution. Through this method, the next generation is expected to have better diversity.

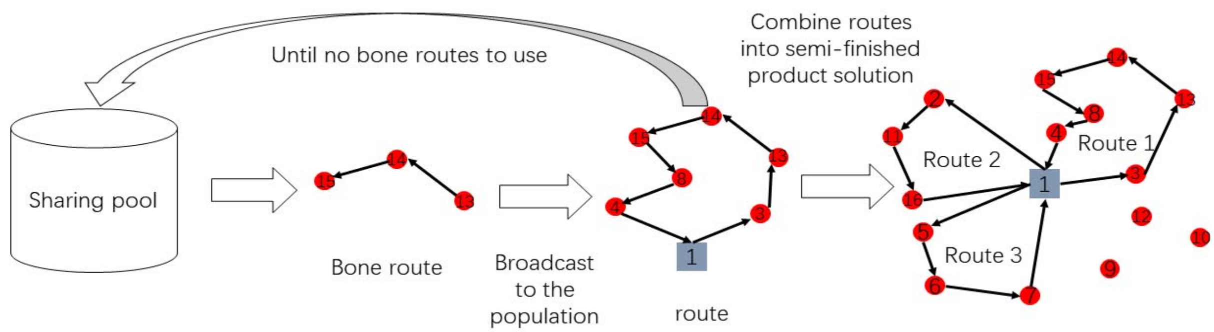

Fig 2. The creation of semi-finished product solution

V. Numerical Experiment

In this section, experimental studies are conducted to verify the proposed algorithm MOMFMA. Particularly, we describe the experimental design and results. MOMFMA is compared to three classical algorithms, i.e., the multi-objective genetic algorithm (MOGA) [15], the multi-factorial memetic algorithm (MFMA) [11] (with the original local search replaced by the large neighborhood local search to solve VRPTW) and MFEA [5]. MOGA is only applied to the first task defined in Eq. (3).

A. Test Instances

In the experiment, all the three algorithms are tested on 56 Solomon's VRPTW instances [16]. All the dataset contain 100 customers. The 56 problems are divided into three types, i.e., C, R, and RC. Each type has two subtypes called 1 and 2. The problems in C category are clustered data, where nodes are clustered either in terms of geographical locations or serve time. The problems in R category are distributed randomly. The problems in RC group are the combinations of C and R problems. The subtype 1 indicates narrower time window for the depot and subtype 2 means wider time window for the depot.

B. Parameter Settings

For a fair comparison, the three algorithms use the same population size N=100 and a maximum of 130000 fitness evaluations. Because of the involvement of local search in MOMFMA, the number of iterations of the three algorithms are different. The random mating probability (*rmp*) is set to 0.3. The crossover rate and mutation rate of the three algorithms are set to 0.5. The coefficients in Eqs. (4) and (7) are set to K_1=0.6, K_2=0.4, K_3=0.5, and a=b=0.5.

C. Result and Analysis

The best results of each algorithm over 10 runs on the 56 benchmark problems are presented in Table I. The *NV%* and *TD%* represent the percentage difference from the optimal solution [7, 17, 18, 19] in terms of number of vehicles and route length, respectively. The distance convergences of all algorithms on problems C101, C201, R101, R201, RC101 and RC201 are also plotted in Fig 3. In order to show the convergence of each comparison algorithm clearly, only the first 100 generations are intercepted in Fig 3. It is observed from the results that:

1) the overall performance of MOMFMA is better than the other three algorithms. Because of the implicit and explicit information sharing, MOMFMA is superior to the other three algorithms in terms of convergence speed and solution quality. The combination of MFEA and MOGA with large neighborhood search and bone route shows outstanding performance.
2) From Fig 3, the convergence of distance demonstrates that the implicit information transfer in MFEA works efficiently compared with MOGA. Moreover, based on implicit information transfer, we add the explicit information share. Compared with MFMA, the explicit information transfer in MOMFMA plays a certain role in the process of algorithm. The components of bone route and semi-finished product solution improve the speed of convergence dramatically. This improvement can be attributed to the stratified sampling selection to maintain the diversity of population, so the quality of repaired bone routes is satisfactory.
3) From Table 1, in the case of 300 iterations, in terms of distance, the difference between most of the results reached by MOMFMA and current optimal results is small, and some results can attain the current optimal results. For the number of vehicles, all the results are able to get the current optimal solutions. In conclusion, MOMFMA not only speeds up the convergence, but also improves the quality of the found solutions. Moreover, the difference of results between MFEA and MOGA shows that the idea of defining a problem into two forms is feasible. Under the framework of MFEA, it can accelerate the main task through the auxiliary task. Compared with MFEA, because of the large neighborhood local search, the performance of MFMA is much better. This shows the key of local search for vehicle routing problem.
4) From the results of the four algorithms in three different types of data, it can be found that the algorithms on random distribution data are worse than the other two types. The more randomness in the distribution of the data, the more diverse the structure of solution. This affects the stability of the algorithms and the final results.

VI. CONCLUSIONS

In this paper, a multi-objective multi-factorial memetic algorithm (MOMFMA) has been proposed to solve VRPTW. The proposed MOMFMA can improve the quality of the obtained solution and the speed of population convergence in MTO. We introduce explicit methods to share information between tasks. The creation of bone route and the use of semi-finished product solution actually speed up the optimization of the algorithm. The success of MOMFMA also indicates the explicit information sharing mechanism can improve the convergence of algorithm.

VII. ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grants 61871272 and 61976143, in part by the Natural Science Foundation of Guangdong Province, China, under Grants 2020A151501479, 2019A1515010869 and 2020A151501946, in part by the Shenzhen Scientific Research and Development Funding Program under Grants JCYJ20190808173617147 and GGFW2018020518310863, in part by the Scientific Research Foundation of Shenzhen University for newly introduced teachers, under Grant 2019048 and 85304/00000247, and in part by the Zhejiang Lab's International Talent Fund for Young Professionals.

REFERENCES

[1] T. Bäck, U. Hammel, and H. P. Schwefel, "Evolutionary computation: Comments on the history and current state", *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 3-17, 1997.

[2] L. Feng, L. Zhou, J. H. Zhong, A. Gupta, Y.-S. Ong, and K. C. Tan, "Evolutionary multitasking via explicit autoencoding", *IEEE Transactions on Cybernetics*, vol. 49, no. 9, pp. 3457-3470, 2018.

[3] G. Yu, T. Chai, and X. Luo, "Multiobjective production planning optimization using hybrid evolutionary algorithms for mineral processing", *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 4, pp. 487-514, 2011.

[4] K. C. Tan, Y. H. Chew, and L. H. Lee, "A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows", *Computational Optimization and Applications*, vol. 34, no. 1, pp. 115, 2006.

[5] A. Gupta, Y. S. Ong, and L. Feng, "Multifactorial evolution: toward evolutionary multitasking", *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 343-357, 2015.

[6] J. Tang, Y. Chen, Z. Deng, Y. Xiang, and C. P. Joy. "A group-based approach to improve multifactorial evolutionary algorithm", *2018 International Joint Conference on Artificial Intelligence*, pp. 3870-3876, 2018.

[7] P. Shaw, "Using constraint programming and local search methods to solve vehicle routing problems", *1998 International Conference on Principles and Practice of Constraint Programming*, pp. 417-431, 1998.

[8] K. C. Tan, L. H. Lee, Q. Zhu, and K. Qu, "Heuristic methods for vehicle routing problem with time windows", *Artificial Intelligence in Engineering*, vol. 15, no. 3, pp. 281-295, 2001..

[9] C. D. Tarantilis, and C. T. Kiranoudis, "BoneRoute: An adaptive memory-based method for effective fleet management", *Annals of Operations Research*, vol. 115, no. 4, pp. 227-241, 2002.

[10] B. Da, A. Gupta, Y. S. Ong, and L. Feng, "Evolutionary multitasking across single and multi-objective formulations for improved problem solving", *2016 IEEE Congress on Evolutionary Computation (CEC)*, 2016.

[11] A. Gupta, Y. S. Ong, L. Feng, and K. C. Tan, "Multiobjective multifactorial optimization in evolutionary multitasking", *IEEE Transactions on Cybernetics*, vol. 47, no. 7, pp. 1652-1665, 2016.

[12] Y. S. Ong and A. Gupta, "Evolutionary multitasking: a computer science view of cognitive multitasking", *Cognitive Computation*, vol. 8, no. 2, pp. 125-142, 2016.

[13] A. Gupta, J. Mańdziuk, and Y. S. Ong, "Evolutionary multitasking in bi-level optimization", *Complex & Intelligent Systems*, vol. 1, no. 4, pp. 83-95, 2015.

[14] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002

[15] C. M. Fonseca, and P. J. Fleming, "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization", *ICGA*, vol. 93, no. July, pp. 416-423, 1993.

[16] M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints", *Operations Research*, vol. 35, no. 2, pp. 254-265, 1987.

[17] N. Kohl and O. B. Madsen, "An optimization algorithm for the vehicle routing problem with time windows based on Lagrangian relaxation", *Operations Research*, vol. 45, no. 3, pp. 395-406, 1997.

[18] S. R. Thangiah, I. H. Osman, and T. Sun, "Hybrid genetic algorithm, simulated annealing and tabu search methods for vehicle routing problems with time windows", *Technical Report SRU CpSc-TR-94-27*, vol. 69, 1994.

[19] S. R. Thangiah, "Vehicle routing with time windows using genetic algorithms", Artificial Intelligence Lab., Slippery Rock Univ, 1993.

[20] A. Gupta, and Y.-S. Ong. "Memetic Computation: The Mainspring of Knowledge Transfer in a Data-Driven Optimization Era", *Springer*, 2018.

[21] A. Gupta, Y.-S. Ong, and L. Feng, "Insights on transfer optimization: Because experience is the best teacher", *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 51-64, 2017.

[22] L. Feng, W. Zhou, L. Zhou, S. Jiang, J. Zhong, B. Da, Z. Zhu, and Y. Wang, "An empirical study of multifactorial PSO and multifactorial DE", *2017 IEEE Congress on Evolutionary Computation*, pp. 921-928, 2017.

[23] Z. Liang, J. Zhang, L. Feng, and Z. Zhu, "A hybrid of genetic transform and hyper-rectangle search strategies for evolutionary multi-tasking," *Expert Systems with Applications*, vol. 138, article no. 117798, 2019.

[24] Y. Yu, A. Zhu, Z. Zhu, Q. Lin, J. Yin and X. Ma, "Multifactorial differential evolution with opposition-based learning for multitasking optimization," *2019 IEEE Congress on Evolutionary Computation*, pp. 1898-1905, 2019.

[25] J. Yin, A. Zhu, Z. Zhu, Y. Yu and X. Ma, "Multifactorial evolutionary algorithm enhanced with cross-task search direction," *2019 IEEE Congress on Evolutionary Computation*, pp. 2244-2251, 2019.

[26] Q. Chen, X. Ma, Z. Zhu, and Y. Sun, "Evolutionary multi-tasking single-objective optimization based on cooperative coevolutionary memetic algorithm," *The 13th International Conference on Computational Intelligence and Security*, pp.197-201, 2017

[27] Q. Chen, X. Ma, Y. Sun, and Z. Zhu, "Adaptive memetic algorithm based evolutionary multi-tasking single-objective optimization,"*The 11th International Conference on Simulated Evolution and Learning*, pp.462-472, 2017.

[28] X. Ma, Q. Chen, Y. Yu, Y. Sun, L. Ma, and Z. Zhu, "A two-level transfer learning algorithm for evolutionary multitasking." *Frontiers in Neuroscience*, vol. 13, article no. 1408, 2020.

TABLE I: THE BEST RESULTS OBTAINED BY THE ALGORITHMS

Problems	Published best		MOMFMA				MFMA		MFEA		MOGA	
	f_1	f_2	f_1	f_2	*NV%*	*TD%*	f_1	f_2	f_1	f_2	f_1	f_2
C101	10	828.94	**10**	**852.32**	0	2.8	11	918.50	22	1838.26	26	2043.54
C102	10	828.94	**10**	**841.98**	0	1.5	10	938.24	19	1721.64	21	1876.72
C103	10	828.06	**10**	**846.23**	0	2.2	10	933.62	15	1536.78	14	1676.42
C104	10	824.78	**10**	**832.29**	0	0.9	10	1053.15	11	1293.38	13	1539.48
C105	10	828.94	**10**	**839.97**	0	1.3	10	905.69	18	1784.07	24	2082.93
C106	10	828.94	**10**	**831.84**	0	0.3	10	923.86	20	1590.33	23	1961.63
C107	10	828.94	**10**	**855.60**	0	3.2	10	903.49	19	1716.83	20	2012.03
C108	10	828.94	**10**	**845.98**	0	2.0	10	899.63	16	1532.6	19	1701.45
C109	10	828.94	**10**	**844.35**	0	1.8	10	892.49	14	1434.89	17	1746.27
C201	3	591.56	**3**	**591.56**	0	0	3	591.56	14	1661.85	17	2006.77
C202	3	591.56	**3**	**591.56**	0	0	3	591.56	11	1556.71	9	1793.94
C203	3	591.17	3	600.17	0	1.5	**3**	**598.31**	8	1407.78	7	1515.88
C204	3	590.60	**3**	**607.82**	0	2.9	3	663.31	6	1233.91	5	1383.61
C205	3	588.88	**3**	**588.88**	0	0	**3**	**588.88**	13	1520.79	15	1770.32
C206	3	588.49	**3**	**588.49**	0	0	**3**	**588.49**	9	1337.05	12	1675.43
C207	3	588.49	**3**	**588.49**	0	0	**3**	**589.49**	9	1446.45	12	1569.36
C208	3	588.32	**3**	**588.32**	0	0	**3**	**588.32**	8	1390.36	11	1573.97
R101	19	1645.79	19	1694.29	0	2.9	19	1674.16	28	2076.59	32	2220.32
R102	17	1486.12	**17**	**1510.88**	0	1.6	17	1560.65	25	1883.37	27	2107.5
R103	13	1292.68	13	1334.12	0	3.2	14	1309.78	20	1629.97	22	1806.71
R104	9	1007.24	**9**	**1040.41**	0	3.2	11	1142.03	15	1418.2	16	1585.66
R105	14	1377.11	**14**	**1401.19**	0	1.7	14	1402.62	25	1835.22	27	2039.05
R106	12	1251.98	**12**	**1258.29**	0	0.5	13	1309.62	21	1714.11	22	1936.31
R107	10	1104.66	**10**	**1139.53**	0	3.1	11	1173.60	18	1531.24	18	1688.42
R108	9	960.88	**9**	**971.54**	0	1.1	11	1106.48	14	1325.39	15	1506.75
R109	11	1194.73	**11**	**1201.40**	0	0.5	12	1285.47	20	1646.97	22	1804.49
R110	10	1118.59	**10**	**1134.89**	0	1.4	10	1198.26	18	1542.71	18	1677.24
R111	10	1096.72	**10**	**1123.67**	0	2.4	11	1180.53	17	1558.63	20	1739.17
R112	9	982.14	**9**	**1001.20**	0	1.9	11	1110.53	15	1325.24	15	1509
R201	4	1252.37	4	1252.37	0	0	4	1252.37	13	1656.17	15	1820.16
R202	3	1191.70	**3**	**1220.25**	0	2.3	3	1191.70	10	1538.08	8	1667.81
R203	3	939.54	**3**	**959.89**	0	2.1	3	984.37	8	1294.99	6	1405.96
R204	2	825.52	**2**	**845.69**	0	2.4	2	849.92	4	1067.46	4	1233.61
R205	3	994.42	**3**	**1019.79**	0	2.5	3	1049.7	9	1379.07	11	1518.46
R206	3	906.14	**3**	**925.36**	0	2.1	3	971.58	6	1276.37	7	1443.31
R207	2	893.33	**2**	**912.98**	0	2.2	3	937.17	5	1173.29	6	1352.64
R208	2	726.75	**2**	**740.33**	0	1.8	3	772.55	4	1032.44	3	1145.84
R209	3	909.16	**3**	**930.54**	0	2.3	3	964.17	8	1259.5	10	1436.71
R210	3	939.34	**3**	**952.12**	0	1.3	3	1008.57	8	1332.73	8	1431.58
R211	2	892.71	**2**	**920.68**	0	3.1	2	892.71	5	1121.73	7	1237.57
RC101	14	1696.94	14	1704.17	0	0.4	14	1696.94	25	2295.65	28	2520.9
RC102	12	1554.75	12	1599.53	0	2.8	12	1554.75	20	2030.25	25	2352.77
RC103	11	1264.67	**11**	**1298.9**	0	2.7	12	1371.1	19	1870.69	20	2052.58
RC104	10	1135.48	**10**	**1157.33**	0	1.9	11	1289.49	16	1738.16	15	1854.99
RC105	13	1629.44	13	1660.14	0	1.8	13	1642.02	24	2233.22	26	2309.18
RC106	11	1424.73	**11**	**1456.12**	0	2.2	12	1466.05	22	2066.28	23	2257.74
RC107	11	1230.48	**11**	**1241.1**	0	0.8	12	1327.18	18	1818.49	21	2072.77
RC108	10	1139.82	**10**	**1150.61**	0	0.9	12	1322.4	16	1677.76	18	1913.07
RC201	4	146.91	4	1413.46	0	0.4	4	1406.91	15	1911.7	17	1857.19
RC202	3	1367.09	**3**	**1367.09**	0	0	3	1367.09	10	1829.64	10	1997.51
RC203	3	1049.62	3	1065.96	0	1.5	3	1049.62	7	1549.29	7	1730.5
RC204	3	798.41	**3**	**820.13**	0	2.7	4	912.49	5	1199.26	5	1434.85
RC205	3	1297.19	3	1315.56	0	1.4	4	1313.47	12	1865.44	10	2108.46
RC206	3	1146.32	**3**	**1175.98**	0	2.6	4	1189.75	9	1704.12	12	1879.52
RC207	3	1061.14	**3**	**1089.66**	0	2.7	4	1106.99	8	1591.37	10	1718.54
RC208	3	828.14	**3**	**842.41**	0	1.7	3	927.25	6	1202	7	1376.72

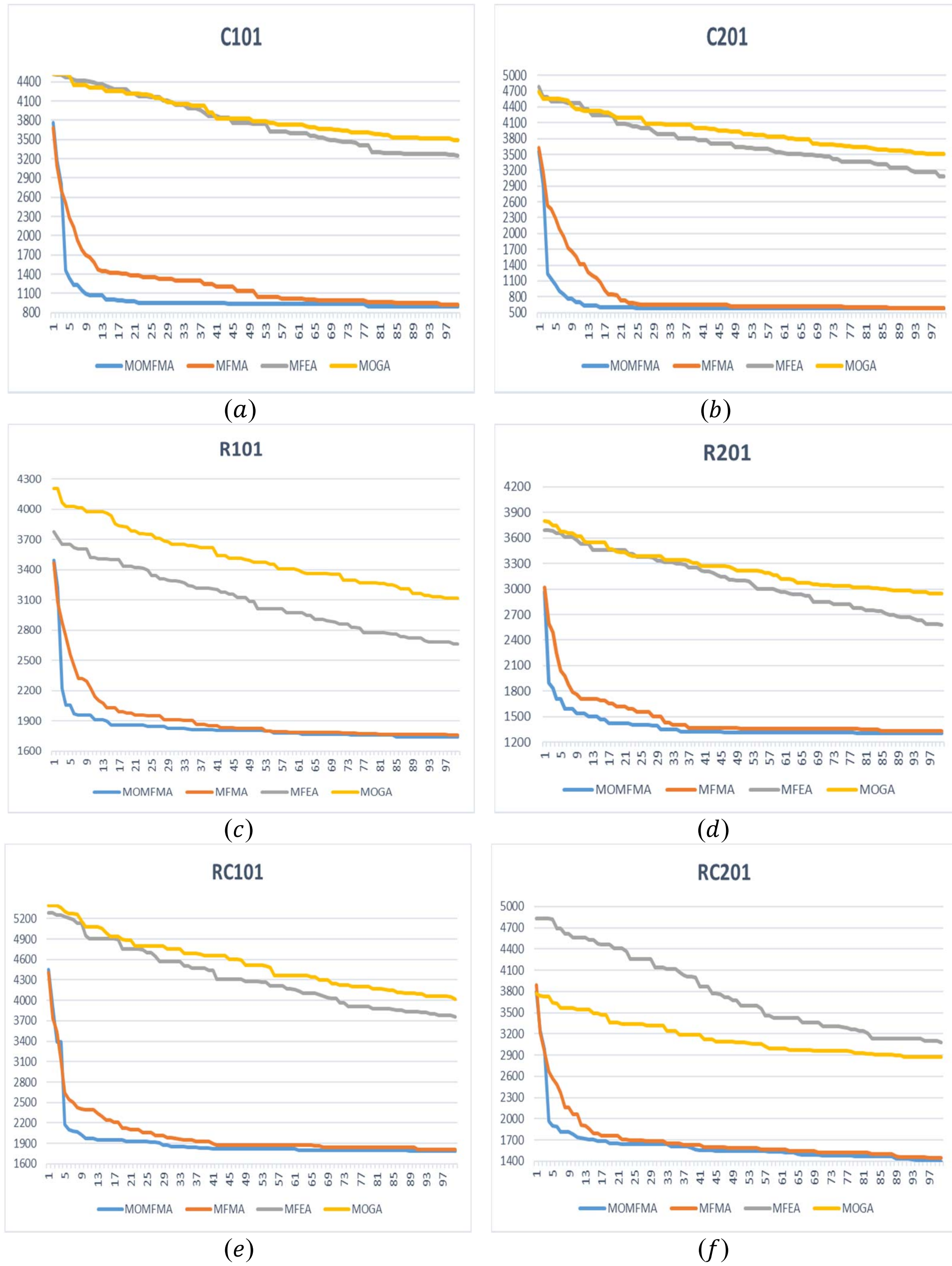

Fig 3. The convergence of distance of the four algorithms