

Multifactorial Evolutionary Algorithm For Clustered Minimum Routing Cost Problem

Tran Ba Trung
School of Information and
Communication Technology, Hanoi
University of Science and Technology
Hanoi, Vietnam
batrung97@gmail.com

Huynh Thi Thanh Binh
School of Information and
Communication Technology, Hanoi
University of Science and Technology
Hanoi, Vietnam
binhht@soict.hust.edu.vn

Le Tien Thanh
School of Information and
Communication Technology, Hanoi
University of Science and Technology
Hanoi, Vietnam
thanhbok26b@gmail.com

Ly Trung Hieu
School of Information and
Communication Technology, Hanoi
University of Science and Technology
Hanoi, Vietnam
lytrunghieu5896@gmail.com

Pham Dinh Thanh
The Faculty of Natural Science and
Technology, Tay Bac University
Sonla, Vietnam
thanhp05@gmail.com

ABSTRACT

Minimum Routing Cost Clustered Tree Problem (CluMRCT) is applied in various fields in both theory and application. Because the CluMRCT is NP-Hard, the approximate approaches are suitable to find the solution for this problem. Recently, Multifactorial Evolutionary Algorithm (MFEA) has emerged as one of the most efficient approximation algorithms to deal with many different kinds of problems. Therefore, this paper studies to apply MFEA for solving CluMRCT problems. In the proposed MFEA, we focus on crossover and mutation operators which create a valid solution of CluMRCT problem in two levels: first level constructs spanning trees for graphs in clusters while the second level builds a spanning tree for connecting among clusters. To reduce the consuming resources, we will also introduce a new method of calculating the cost of CluMRCT solution. The proposed algorithm is experimented on numerous types of datasets. The experimental results demonstrate the effectiveness of the proposed algorithm, partially on large instances.

CCS CONCEPTS

• **Mathematics of computing** → **Approximation algorithms; Discrete optimization.**

KEYWORDS

Clustered Minimum Routing Cost Problem, Clustered Tree Problem, Multifactorial Evolutionary Algorithm, Evolutionary Algorithm

ACM Reference Format:

Tran Ba Trung, Huynh Thi Thanh Binh, Le Tien Thanh, Ly Trung Hieu, and Pham Dinh Thanh. 2019. Multifactorial Evolutionary Algorithm For

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

SoICT 2019, December 4–6, 2019, Hanoi - Ha Long Bay, Viet Nam

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-7245-9/19/12...\$15.00

<https://doi.org/10.1145/3368926.3369712>

Clustered Minimum Routing Cost Problem. In *The Tenth International Symposium on Information and Communication Technology (SoICT 2019), December 4–6, 2019, Hanoi - Ha Long Bay, Viet Nam*. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3368926.3369712>

1 INTRODUCTION

The class of problems related to finding minimum cost clustered trees is among the most widely studied problems in applied mathematics and theoretical computer science. In general, clustered tree problems have many applications in the field of computer network design, computational biology, transportation and logistics as well as water resource management [12, 13]. Numerous problems with different cost function has received much attention such as Clustered Minimum Steiner Tree [21], Clustered Shortest-Path Tree [7] and Minimum Routing Cost Clustered Tree Problem (CluMRCT) [12].

Among those mentioned problems, CluMRCT is one of the most newly investigated problems. This problem has been formally formulated in [12]. Concretely, consider a connected, undirected graph $G = (V, E, w)$ with nonnegative edge length function $w : E \rightarrow \mathbb{R}^+$. The vertices V are partitioned into k clusters $R = \{R_1, R_2, \dots, R_k\}$. A spanning tree T of G is a clustered spanning tree if T can be cut into k subtrees by eliminating $k - 1$ edges such that each subtree is a spanning tree for only one cluster. The CluMRCT problem focuses on the routing cost, which is the sum of shortest path distance between any pairs of vertices given a clustered spanning tree T . The CluMRCT problem is finding on graph G a clustered spanning tree T having the minimum routing cost. One practical realization of this problem is in computer network application where the communication terminals are vertices, and they are partitioned into many clusters. The communication between those terminals are restricted within a cluster and only a few terminals can be connected to another cluster for maximizing efficiency and other security concerns. Solving CluMRCT is equivalent to facilitate the network architecture that consumes the minimum peer-to-peer communication resources.

The CluMRCT has been proven to be NP-Hard by showing its relationship with an equivalent problem named clustered *st*-path problem, which is also proven to be NP-Hard [12]. However, it is a pressing problem and approximate solutions are acceptable in practice. There are various approaches for finding approximate solutions for these type of intractable problem such as approximation algorithms, heuristic algorithms or meta-heuristic algorithms. However, methods to solve CluMRCT are limited to only a 2-approximation algorithm for the case of three clusters [12]. In the practical application of CluMRCT for computer network, there would be multiple designs of the computer network topologies with various ways to set up the clusters constraint, thereby arising the problem of solving not only one CluMRCT but multiple CluMRCT instances. These instances do not exist in isolation; thus, effectively sharing the underlying pattern of minimum cost clustered tree for one instance may result in a better solution in another instance.

Motivated by that practical indication, this paper aims to design a Multifactorial Evolutionary Algorithm (MFEA) [9] to simultaneously solve multiple minimum cost clustered trees problem instances together. MFEA is a variant of Evolutionary Algorithm (EA) [15, 17], which is classified as a meta-heuristic algorithm that capable to provide a near-optimal solution for those mentioned hard problems in an acceptable timing limit. It is also being an algorithmic realization of a widely-study multitasking optimization scheme. It is attracting considerable interest because of its capability to optimize and facilitate better solutions for multiple related problems. The driving force behind MFEA is its genetic transfer using crossover across different tasks that share a partial proportion of high-quality solution from one task to another [8]. If problems optimized by MFEA having the highly correlated fitness landscape and near global optima, the high-quality solution from one task will definitely having high performance in another, leading to significant improvement in every constitutive task [1, 10].

In particular, our contributions can be summarized as follow:

- We proposed a novel evolutionary algorithm to solve the CluMRCT, including a two levels meta-heuristic crossover and mutation operators on adjacency list representation of the problem.
- By following the generic procedure of MFEA, we expanded the representation and operators to facilitate a common platform for facilitating knowledge exchange between different instances of CluMRCT of the various number of clusters and cluster size.

This paper is organized as follows. Section 2 again reformulates the CluMRCT and presents comprehensive details on the way to efficiently compute its cost function. Section 3 examines the literature of the considered problem and the development of MFEA. The next section provides the two proposed algorithms. An empirical study to analyze the effectiveness of the proposed approaches is provided in Section 5. A discussion of the result and future direction is presented in Section 6

2 PROBLEM FORMULATION

Given a weighted undirected graph $G = (V, E, w)$. V and E are the vertex and the edge sets, respectively. In particular, all paths in the graph G are simple path. A path in G is simple if no vertex appears

more than once on it. w is a nonnegative edge length function $w : E \rightarrow \mathbb{R}^+$. An edge between vertices u and v , where $u, v \in V$, is denoted by (u, v) . The weight of edge (u, v) is denoted by $w(u, v)$. A graph G is connected if all pairs of vertex within the graph is connected. A pair of vertices (u, v) is connected if G contains a path from u to v . For a graph G , $V(G)$ and $E(G)$ denote the vertex and the edge sets, respectively. For a vertex subset U , the subgraph of G induced by U is denoted by $G[U]$.

The authors define the H-graph $G' = (V', E')$ where each vertex represents a cluster R_i . An edge connecting two vertices of G' exists if the two corresponding clusters in G are connected by at least one edge. A spanning tree T of the graph G is a clustered spanning tree if all the vertices in the same cluster R_i are clustered together in T . It means that T can be cut into k subtrees by removing $k - 1$ edges such that each subtree is a spanning tree for one cluster R_i .

The objective of CluMRCT problem mainly focuses on the routing cost $c(T)$. The routing cost $c(T)$ is the total distance over all pairs of vertices in the connecting path defined by the clustered spanning tree T :

$$c(T) = \sum_{u, v \in V(T)} d_T(u, v) \quad (1)$$

where $d_T(u, v)$ is the distance between u and v on T . The CluMRCT problem searches for the clustered spanning tree T that minimize the routing cost:

$$\underset{T}{\text{minimize}} \ c(T) = \sum_{u, v \in V(T)} d_T(u, v) \quad (2)$$

3 RELATED WORK

In recent years, there has been considerable interest in solving the minimum cost clustered tree problem and its variants [12]. Major developments have been carried out in developing approximate algorithms for solving the NP-hard Clustered Steiner Tree Problem (CluSteiner) variant and deriving the Steiner ratio of the problem [21]. Chen subsequently proposed another approximate algorithm that improved the performance ratio by 40% under the condition of distance metric satisfying the triangular inequality. Another NP-hard variant of the general class of minimum cost clustered tree problem called Clustered Shortest-Path Tree Problem (CluSPT) has also been widely investigated in recent literature, under theoretical viewpoint [7] as well as developing approximation algorithms for this problem [4, 19]. Following studies from Thanh et al. suggested another approach for solving this class problem more effectively by adopting the evolutionary multitasking paradigm [18, 20].

CluMRCT that we are focusing on this study is also a typical variant of the general minimum-cost clustered tree that having numerous practical applications. For instance, applications of CluMRCT problem arise in the fields of network design, computational biology and transportation [12]. CluMRCT is proven NP-hard by a transformation to the equivalent problem of. Because of its NP-hardness, a popular method to tackling the CluMRCT would be developing approximation algorithm, heuristic or meta-heuristic algorithm. However, to the best of our knowledge, there are no published study has proposed such method for this problem.

In recent literature, MFEA [3] is a newly founded meta-heuristic algorithm that has been gaining much attention due to its ability to solve multiple hard problems more effectively than the traditional EA [9]. Some preliminary work was carried out suggesting that MFEA is being able to capitalize the knowledge overlap across relevant tasks to facilitate better objective value on solving multiple combinatorial problems concurrently [11, 22]. Other studies attempt to investigate the theoretical foundation of MFEA as well as improve this algorithm such that its performance would not be impeded by solving tasks that are two conflicting in their fitness landscape and global optima [2, 5]. Also, there is no prior attempt to design a MFEA for solving CluMRCT.

4 PROPOSED ALGORITHM

In this section, we introduce MFEA to solve multiple CluMRCT problems simultaneously. A CluMRCT problem corresponding to a particular task in MFEA. Tasks are different number of clusters and different dimension is encoded into a unified individual. Each unified individual is divided into two-level, the first deal with constructing the spanning tree inside the clusters, the second problems deal with constructing tree connect between clusters. In the entire optimization process, we introduce novel two-levels genetic operators to construct spanning tree starting from the smallest task to the biggest task. As a result, the unified individual is the biggest tree contains the solution of all tasks.

4.1 Individual encoding, Population Initialization and Decoding method

An individual in the unified search space encompassing the genetic material of all of the tasks. An individual in the unified search space consists of the different solutions of CluMRCT problems corresponding to tasks. An individual is a spanning tree which is the number of clusters is the maximum number of the clusters from all of the tasks, and the cluster inside takes the maximum number of vertices from the all of all tasks. The clusters in each instance of CluMRCT problem is ascending sorted by the number of vertices before encoding into a unified individual to maximize the number of overlapping dimensions of different tasks. As a result, a unified individual captures solution for all task and the decoding method only finds the corresponding sub-graph with the task to produce valid solution for particular CluMRCT task. After defining structures of an individual, we generate a new individual for the CluMRCT problem. Initial individual is constructed randomly by applying PrimRST algorithms in the smallest task, all initial bigger tasks are generated from the smaller task and input graph by detecting and delete cycle until gaining spanning tree for the biggest task. The detail about encoding, decoding, and initializing individual are presented in [20].

4.2 Crossover operator

We introduce a new crossover operator for individuals in unified search space. In particular, the novel crossover operator generates two offsprings from two parents. Two offsprings are constructed from the set of edges of two parents. The crossover operator starts by selecting two individuals in unified search space from the current population to determine the common edges belong to each

Algorithm 1: Proposed crossover operator

Input: Two parents: $T_t = (V, E_t, C)$, $t = 1, 2$;
Output: Two offspring $S_1 = (V_s, E_{s_1})$ and $S_2 = (V_s, E_{s_2})$;

```

1 begin
2    $V_s \leftarrow V$ ;  $\theta \leftarrow |C|$ ;
3   foreach cluster  $j^{th}$  do  $S_1^j, S_2^j \leftarrow DSTX(T_1, T_2)$ ;
4   Determine two H-Graphs  $G'_1$  and  $G'_2$  from  $T_1$  and  $T_2$ ,
   respectively;
5    $T'_1, T'_2 \leftarrow DSTX(G'_1, G'_2)$ ;
6    $E_{s_1} \leftarrow (\cup_{j=1}^{\theta} E(S_1^j)) \cup E(T'_1)$ ;  $E_{s_2} \leftarrow (\cup_{j=1}^{\theta} E(S_2^j)) \cup E(T'_2)$ ;
7   return  $S_1, S_2$ 
8 end

```

individual, the length of this set represents the similarity of two parents. If two individuals have more common edge meaning that two individual close similarities. The algorithm determines the set of edges the only belong to a particular parent.

After that, a new offspring is constructed based on the combination of a set of common edges and a set of edges only belong to its parent. The crossover operator is process in two-levels: the first is applying for spanning inside cluster and the seconds one is applying for spanning tree which is connect among clusters. The details as presenting in Algorithm 1.

The steps of the DSTX algorithm is presented in Algorithm 2. DSTX generates two spanning trees S_1, S_2 . Line 2, 3 define two sets of edges only belong to a particular parent. Line 8 \rightarrow 23 generates two offsprings spanning tree from the smallest task to the biggest task. Line 9 \rightarrow 15 to generate offspring S_1 . In each particular task, the algorithm detects the cycles which are created by adding new edges to parents solution, then delete an edge in each cycle path to obtain the new solution. Removing an edge will be created a new valid solution for the current task but without disrupting the solution for the smaller task.

The Figure 1 illustrates the crossover operator on two parents. The Figure 1(b-d) presents crossover for each cluster, other figures present crossover steps for H-Graphs. Figure 1(b) depicts the sub-graph when combine the sub-graph on clusters in Parent 1 and the set of edges E'_2 only belong to Parent 2. The red dash edges in Figure 1(d) belong to parent 2 obtained by performing the DSTX. Figure 1(e) illustrates the sub-graph after building the spanning tree for clusters. Figure 1(d) and Figure 1(g) show graph after combining H-Graphs in second stages. The offspring obtained by crossover operator is depicted in Figure 1(h) after combining two graph in Figure 1(d) and Figure 1(g).

4.3 Mutation operator

This mutation operator to generate new individual based on applying small variation on the original individual. The main idea of the proposed mutation operator (PMO) is that the PMO adds an edge in the input graph G but not in the individual to create a cycle and then removes an edge in the cycle to create a new spanning tree. The details of the MPO is presented in Algorithm 3.

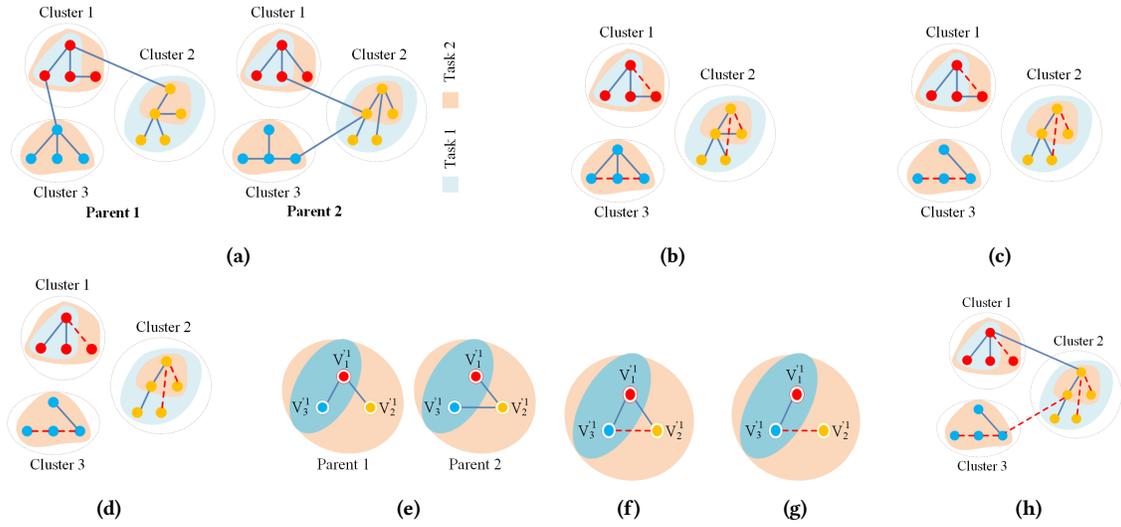


Figure 1: An example of crossover process applying two parents for MFEA with two tasks

Algorithm 2: DSTX

Input: Two spanning trees $T_1 = (V, E_1)$ and $T_2 = (V, E_2)$
 where $V = V_h \supseteq V_{h-1} \supseteq \dots \supseteq V_1$
Output: Two spanning tree $S_1 = (V, E_{s_1})$ and $S_2 = (V, E_{s_2})$

```

1 begin
2    $E'_1 \leftarrow E_1 \setminus E_2$ ;    $E'_2 \leftarrow E_2 \setminus E_1$ ;    $t \leftarrow |E'_1|$ ;
3    $k \leftarrow$  Generate a random number  $k$  from  $[1, t]$ ;
4    $E_{s_1} \leftarrow$  Add  $k$  random edges from  $E'_1$  to  $T_2$ ;
5    $E_{s_2} \leftarrow$  Add  $k$  random edges from  $E'_2$  to  $T_1$ ;
6   for  $i \leftarrow 1$  to  $h$  do
7     foreach edge  $e = (v_1, v_l)$  in  $E'_1$  do
8       if  $v_1, v_l \in V_i$  then
9         Find the path  $P = (v_1, v_2, \dots, v_l)$  between  $v_1$ 
10        and  $v_l$  in the set of edges  $E(T_1) - \{e\}$ ;
11        Choose a vertex  $v_j \in P$  such that  $v_j \in V_i \setminus V_{i-1}$ ;
12        Delete edge  $(v_j, v_{j+1})$  from  $S_1$ ;
13      end
14    end
15    foreach edge  $e = (v_1, v_l)$  in  $E'_2$  do
16      if  $v_1, v_l \in V_i$  then
17        Find the path  $P = (v_1, v_2, \dots, v_l)$  between  $v_1$ 
18        and  $v_l$  in the set of edges  $E(T_2) - \{e\}$ ;
19        Choose a vertex  $v_j \in P$  such that  $v_j \in V_i \setminus V_{i-1}$ ;
20        Delete edge  $(v_j, v_{j+1})$  from  $S_2$ ;
21      end
22    end
23  end
  
```

Algorithm 3: Proposed mutation operator

Input: Input graph $G = (V_G, E_G, C_G)$ where
 $C_G = C^1 \cup C^2 \cup \dots \cup C^\theta$, $C^p \cap C^q = \emptyset$, $\forall p \neq q$; An
 individual $T = (V, E, C)$

Output: A Tree $T' = (V', E')$

```

1 begin
2    $T' \leftarrow T$ ;   Select a random cluster  $C^j$ ;
3    $e = (u, v) \leftarrow$  Select randomly an edge in  $E(G[C^j])$  but not
4   in  $E(T'[C^j])$ ;
5    $P_T(u, v) \leftarrow$  Path between vertex  $u$  and vertex  $v$  in  $T'$ ;
6    $e' \leftarrow$  Select randomly an edge in  $P_T(u, v)$ ;
7    $E(T') \leftarrow (E(T) - \{e\}) \cup \{e'\}$   $\triangleright$  exchange edges  $e$  and  $e'$ ;
8   return  $T'$ ;
9 end
  
```

4.4 Evaluation method

In this sub-section, we present a method to calculate the cost of CluMRCT solution in $O(n)$ time complexity. The cost function of CluMRCT is the sum of connectivity cost between all pairs of vertices in the graph. The solution of CluMRCT is spanning tree, thus, traveling all vertices in $O(n)$ time complexity and removing an edge from spanning tree will create two connected components. Given $T = (E, V)$ is the CluMRCT solution, the cost of two adjacency vertices on solution T are calculated as the following formula $C(v_1, v_2) = |V_1| * |V_2| * w(v_1, v_2)$ where $|V_1|$ is the number of vertices in connected component contains v_1 after remove $e = (v_1, v_2)$ from T . $|V_2|$ is the number of vertices in connected component contains v_2 after remove $e = (v_1, v_2)$ from T . $w(v_1, v_2)$ is connecting cost between two adjacency vertices v_1 and v_2 .

The cost of CluMRCT solution T is calculated as following four-step procedure:

- **Step 1:** Assign a random vertex as start vertex s .

- **Step 2:** BFS algorithms start with s vertex then store the order of visiting to the visited list.
- **Step 3:** The cost of two adjacency vertices on solution T are calculated as following: Travel bottom-up (leaves to root) in visited list to update the label of current vertex to $l = d + 1$, where d is the number of descendant vertices of current vertex. Update the label of ancestor vertex to $l_a = l_a + l$. The cost between current vertex v_1 and its ancestor v_2 is calculated by formula $C(v_1, v_2) = l * (n - l) * w(v_1, v_2)$. Where, l is the current vertex label, n is the number of vertices, $w(v_1, v_2)$ is connecting cost between current vertex v_1 and its ancestor v_2 .
- **Step 4:** The cost function of CluMRCT is calculated follow as: $\sum_{v_1 \in V} \sum_{v_2 \in V} C(v_1, v_2)$. Where, $C(v_1, v_2)$ is the cost of two adjacency v_1 and v_2 in CluMRCT solution T .

5 EXPERIMENTAL RESULTS

5.1 Problem instances

To the best of our knowledge, there is no publicly available set of benchmark for the CluMRCT problem. Therefore, we utilized a set public test instances for an equivalent clustered tree problem which is the CluSPT dataset version 3 on Mendeley [16]. This dataset included six distinct types of instances that were generated by various algorithms [4] and categorized into two types conformable to their dimension. The instances were appropriate for evaluating cluster problems [14]. However, to test the effectiveness of proposed algorithms to solving the CluMRCT instead of CluSPT, we ignore the information of the global source vertex of this CluSPT dataset. For evaluation of the proposed algorithms, instances with dimensionality from 30 to 500 were selected.

5.2 Experimental setup

To evaluate the performance of the proposed algorithm for the CluMRCT, the authors implemented the metaheuristic algorithm EA and MFEA. Then, the results of two algorithms for each instance were compared with each other in term of the best and average fitness of the optimized solutions.

Each scenario was simulated for 30 times on the computer (Intel Core i7 4790, 16GB RAM), with a population size of 100 individuals evolving through 500 generations. The random mating probability is 0.5 and the mutation rate is 0.05. The source codes were installed by Java programming language.

5.3 Experimental criteria

Criteria for evaluating the quality of the result of the algorithms are presented in the following table:

Criteria	Description
Average (Avg)	Average function value over all runs
Best-found (BF)	Best function value achieved over all runs
Relative Percentage Differences (RPD)	The difference between the average costs of two algorithms

In order to compare the quality of the CluMRCT solutions received from EA and MFEA, RPD is used to compute the difference between the average costs of two algorithms. The RPD is calculated

by the following formula: $RPD(A, B) = \frac{C_B - C_A}{C_B} * 100\%$ where C_B is the average cost of a solution obtained from EA, C_A is the average cost of a solution obtained from MFEA.

5.4 Experimental results

The experimental results show that the quality of the CluMRCT solution of MFEA algorithm is better than EA algorithm in most scenario. Particularly, MFEA's results are better than EA's results in all instances in Type 3, Type 4, Type 1 Large, Type 5 Large and Type 6 Large. It means that the larger instance's size, the more MFEA tends to outperform than EA. There is a noteworthy point that with large instances, the results obtained by MFEA are always better than one obtained by EA in both average result and best-found in any Type. The comparison of results obtained by EA and results obtained by the proposed algorithm is presented in detail in the Table 1.

Table 1: The summarized results demonstrate the proportion of instances that MFEA outperformed EA in term of number of instances having higher best-found fitness and average fitness

	Type 1		Type 3	Type 4	Type 5		Type 6	
	Small	Large	Large	Large	Small	Large	Small	Large
BF	22	10	5	8	15	14	33	13
Avg	20	10	5	8	15	14	33	13
Total	24	10	5	8	16	14	34	13

The details results obtained by algorithms are presented in Table 2 – Table 5. In this table, the italic red cells on a column to show that on those instances, this algorithm outperforms than the other algorithm. The biggest RPD(MFEA, EA) is 98.59% while the biggest RPD(EA, MFEA) is 16.9.

5.5 Convergence trend

The functions in [9] was used for computing the normalized and averaged normalized objectives for analyzing the proposed MFEA algorithm's convergence trends.

The convergence trend during the initial stages of the multi-tasks is depicted in Figure 2(a) for instances 10berlin52 and 10eil51 in Type 1; instances 4berlin52-2x2 and 42rat99-6x7 in Type 6. In this figure, multi-tasks (MT) converges faster than Single task (ST) at any time in the test instances of Type 1. With Type 6's instances, ST converges faster than MT at the begin but MT has gradually surpassed ST from evaluation 100 to the end. In summary, the MT performance suggests better overall convergence characteristics than the ST performance.

With the same instances and genetic operators were used in ST and MT, the improvement can be reached because of the exploitation of multiple function landscapes via implicit genetic transfer, such as the evolutionary multitasking paradigm affords.

Refer to Figure 2(b) and Figure 2(c) for a better understanding of the improved performance because of MT. The figure depicts the convergence trends corresponding to each constitutive task. As you can see, in most instances, the convergence of MT is faster in comparison with ST as instances 10berlin52 and 10eil51 in Figure 2(b). But, some case, like 4berlin52-2x2 in Figures 2(c), the convergence

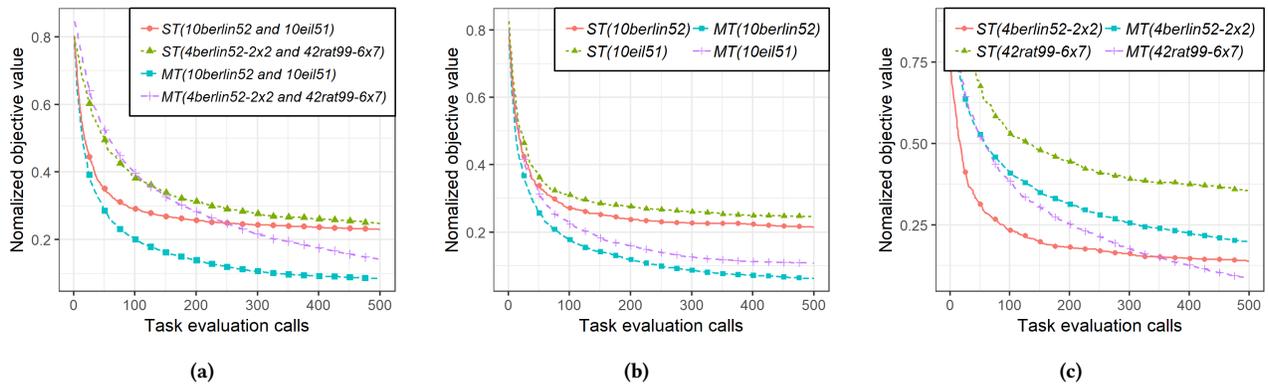


Figure 2: Comparing convergence trends of \tilde{f} , \tilde{f}_1 and \tilde{f}_2 in multi-tasks and serial single task for instances in Types 1, 6.

of MT was slower than the one of ST. In this case, 4berlin52-2x2 and 42rat99-6x7 are simultaneously solved, but two instances are very different structure of solution. 4berlin52-2x2 has 4 clusters and 52 vertices, meanwhile 42rat99-6x7 has 42 clusters and 99 vertices. In unified search space, the shared representation of two solutions is very small, only 4 clusters 42rat99-6x7 impact to 4berlin52-2x2. As a result, the two instances are less impacting each other in process of transferring knowledge during multitasking is the cause of ST outperform than MT.

6 CONCLUSION AND DISCUSSION

This paper proposed a multitask optimization algorithm in the realm of the general MFEA to solve the multiple instances of CluMRC T problem together. Evolutionary operators and a new method for reducing consuming resource of evaluating the solution are also described. The proposed multitasking algorithm is tested on many datasets. Experimental results showed that the proposed MFEA outperforms its single-task variant in most test cases.

REFERENCES

- [1] K. K. Bali, A. Gupta, L. Feng, Y. S. Ong, and Tan Puay Siew. 2017. Linearized domain adaptation in evolutionary multitasking. In *2017 IEEE Congress on Evolutionary Computation (CEC)*. Institute of Electrical and Electronics Engineers (IEEE), Donostia/San Sebastian, Spain., 1295–1302.
- [2] K. K. Bali, Y. Ong, A. Gupta, and P. S. Tan. 2019. Multifactorial Evolutionary Algorithm with Online Transfer Parameter Estimation: MFEA-II. *IEEE Transactions on Evolutionary Computation* (Early Access), - (march 2019), 1–1.
- [3] Kavitesh Kumar Bali, Yew-Soon Ong, Abhishek Gupta, and Puay Siew Tan. 2019. Multifactorial Evolutionary Algorithm with Online Transfer Parameter Estimation: MFEA-II. *IEEE Transactions on Evolutionary Computation* (2019).
- [4] Huynh Thi Thanh Binh, Pham Dinh Thanh, and Ta Bao Thang. 2019. New approach to solving the clustered shortest-path tree problem based on reducing the search space of evolutionary algorithm. *Knowledge-Based Systems* 180 (2019), 12 – 25.
- [5] Y. Chen, J. Zhong, L. Feng, and J. Zhang. 2019. An Adaptive Archive-Based Evolutionary Framework for Many-Task Optimization. *IEEE Transactions on Emerging Topics in Computational Intelligence* (Early Access), - (june 2019), 1–16.
- [6] Yen Hung Chen. 2017. The Clustered and Bottleneck Clustered Selected-Internal Steiner Tree Problems. In *The Second Malta Conference in Graph Theory and Combinatorics 2017*. The Second Malta Conference in Graph Theory and Combinatorics, Qawra, St Paul’s Bay, 44.
- [7] Mattia D’Emidio, Luca Forlizzi, Daniele Frigioni, Stefano Leucci, and Guido Proietti. 2019. Hardness, approximability, and fixed-parameter tractability of the clustered shortest-path tree problem. *Journal of Combinatorial Optimization* 38, 1 (01 Jul 2019), 165–184.
- [8] A. Gupta and Y. Ong. 2016. Genetic transfer or population diversification? Deciphering the secret ingredients of evolutionary multitask optimization. In

- 2016 *IEEE Symposium Series on Computational Intelligence (SSCI)*. Institute of Electrical and Electronics Engineers (IEEE), Athens, Greece, 1–7.
- [9] A. Gupta, Y. Ong, and L. Feng. 2016. Multifactorial Evolution: Toward Evolutionary Multitasking. *IEEE Transactions on Evolutionary Computation* 20, 3 (June 2016), 343–357.
- [10] A. Gupta, Y. S. Ong, B. Da, L. Feng, and S. D. Handoko. 2016. Landscape synergy in evolutionary multitasking. In *2016 IEEE Congress on Evolutionary Computation (CEC)*. Institute of Electrical and Electronics Engineers (IEEE), Vancouver, BC, Canada, 3076–3083.
- [11] Lei Zhou, L. Feng, Jinghui Zhong, Y. Ong, Z. Zhu, and E. Sha. 2016. Evolutionary multitasking in combinatorial search spaces: A case study in capacitated vehicle routing problem. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. Institute of Electrical and Electronics Engineers (IEEE), Athens, Greece, 1–8.
- [12] Chen-Wan Lin and Bang Ye Wu. 2017. On the minimum routing cost clustered tree problem. *Journal of Combinatorial Optimization* 33, 3 (01 Apr 2017), 1106–1121.
- [13] Adriano Masone, Maria Elena Nenni, Antonio Sforza, and Claudio Sterle. 2019. The Minimum Routing Cost Tree Problem. *Soft Computing* 23, 9 (01 May 2019), 2947–2957.
- [14] Mario Mestria, Luiz Satoru Ochi, and Simone de Lima Martins. 2013. GRASP with path relinking for the symmetric Euclidean clustered traveling salesman problem. *Computers and Operations Research* 40, 12 (2013), 3218 – 3229.
- [15] Dinh Thanh Pham and Thi Thanh Binh Huynh. 2015. An effective combination of genetic algorithms and the variable neighborhood search for solving travelling salesman problem. In *2015 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*. IEEE, 142–149.
- [16] Pham Dinh Thanh. 2019. CluSPT Instances. <https://doi.org/10.17632/b4gcgybvt6.3>
- [17] Pham Dinh Thanh, Huynh Thi Thanh Binh, and Bui Thu Lam. 2015. New mechanism of combination crossover operators in genetic algorithm for solving the traveling salesman problem. In *Knowledge and Systems Engineering*. Springer, 367–379.
- [18] P. D. Thanh, D. A. Dung, T. N. Tien, and H. T. T. Binh. 2018. An Effective Representation Scheme in Multifactorial Evolutionary Algorithm for Solving Cluster Shortest-Path Tree Problem. In *2018 IEEE Congress on Evolutionary Computation (CEC)*. Institute of Electrical and Electronics Engineers (IEEE), Rio de Janeiro, Brazil, 1–8.
- [19] P. D. Thanh, H. Thi Thanh Binh, D. D. Dac, N. Binh Long, and L. M. Hai Phong. 2019. A Heuristic Based on Randomized Greedy Algorithms for the Clustered Shortest-Path Tree Problem. In *2019 IEEE Congress on Evolutionary Computation (CEC)*. Institute of Electrical and Electronics Engineers (IEEE), Wellington, New Zealand, 2915–2922.
- [20] H. Thi Thanh Binh, P. Dinh Thanh, T. Ba Trung, and L. Phuong Thao. 2018. Effective Multifactorial Evolutionary Algorithm for Solving the Cluster Shortest Path Tree Problem. In *2018 IEEE Congress on Evolutionary Computation (CEC)*. Institute of Electrical and Electronics Engineers (IEEE), Rio de Janeiro, Brazil, 1–8.
- [21] Bang Ye Wu and Chen-Wan Lin. 2015. On the clustered Steiner tree problem. *Journal of Combinatorial Optimization* 30, 2 (01 Aug 2015), 370–386.
- [22] Y. Yuan, Y. Ong, A. Gupta, P. S. Tan, and H. Xu. 2016. Evolutionary multitasking in permutation-based combinatorial optimization problems: Realization with TSP, QAP, LOP, and JSP. In *2016 IEEE Region 10 Conference (TENCON)*. Institute of Electrical and Electronics Engineers (IEEE), Marina Bay Sands, Singapore, 3157–3164.

Table 2: Details of best-found and average fitness per each instance of Type-3 CluMRCT and Type-4 CluMRCT problem solved by EA and MFEA. The best value are highlighted in boldface.

Instances	EA			MFEA		
	BF	Avg	Avg	BF	Avg	Avg
6i300	7218321	7537212.2	6892856	7139417.6		
6i350	10186959	10979308.4	9528108	10058407.1		
6i400	13838525	14587641.4	13052432	14005791.9		
6i450	17880773	19032256.4	16420320	17838608.4		
6i500	23276800	24761896	20823767	22667302.4		
4i200a	58062080	64788091.5	49873704	57749914.3		
4i200x1	35156736	39884307.3	30221411	36285931.8		
4i200z	35580760	39847121.9	31923515	37291083.8		
4i400a	348050079	381840774.9	235345395	273783937.4		
4i400h	203858230	224087266.5	179831484	204974618.1		
4i400x1	204931109	223854804.4	178575229	202321240.4		
4i400x2	211600994	224658063.9	184342319	204305625.5		
4i400z	189307036	221169556.9	174142050	202083877.8		

Table 3: Details of best-found and average fitness per each instance of Type-5 CluMRCT problem solved by EA and MFEA. The best value are highlighted in boldface.

Instances	EA			MFEA		
	BF	Avg	Avg	BF	Avg	Avg
10i120-46	8620633.2	9393459.8	7784832.4	8197268.9		
10i45-18	875824.2	932734.6	756198.1	851733.6		
10i60-21	1798836.1	2069872.8	1598921.1	1853126.2		
10i65-21	2379511.4	2491125.7	1996599.7	2139469.2		
10i70-21	2630883.1	2873851.8	2324482.7	2546084.3		
10i90-33	3889081.6	4150252.1	3135916.2	3504309.9		
5i120-46	7549373.0	8134562.4	6244382.8	6933990.9		
5i30-17	340097.8	358611.6	343481.4	368298.3		
5i45-18	750733.1	802806.8	658041.4	714002.8		
5i60-21	1771839.1	1974753.5	1581756.3	1746774.6		
5i65-21	2132123.3	2311904.2	1948779.0	2142993.9		
5i70-21	2580773.5	2749384.8	2135118.7	2410433.1		
5i75-22	2740946.6	3012711	2527119.8	2760425.4		
5i90-33	3604939.2	3971655.3	3187292.2	3407267.2		
7i60-21	1960693.7	2139328.6	1637152.6	1848278.6		

Instances	EA			MFEA		
	BF	Avg	Avg	BF	Avg	Avg
10i300-109	47618138.3	4995372.1	40551819.0	45274502.3		
10i400-206	108421226.4	115141165.4	88310142.6	99012666		
10i500-305	182614330.7	198554467.7	138313818.7	145731966.4		
15i300-110	49192926.5	54123296.2	43843232.5	47318804.9		
15i400-207	102953354.8	110503351.8	95421482.9	103134953.9		
15i500-306	176477520.4	185003609.4	156507079.7	164366536.1		
20i300-111	60901743.1	64793312.9	53572569.6	59089047.3		
20i400-208	112865091.8	120286857	99758157.6	105914056.3		
20i500-307	157448014.7	164840517.5	114601072.8	122660401.5		
25i300-112	57643197.6	61742234	49921091.2	52990412.1		
25i400-209	115888694.2	123273581.5	101778949.6	110739520.8		
25i500-308	174495737.9	186917007.2	148853209.8	162658959.2		
5i300-108	59885205.6	64875737.1	567322257.9	61339474.6		
5i400-205	86596796.4	92361135.8	72401881.2	79314034.6		

Table 4: Details of best-found and average fitness per each instance of Type-6 CluMRCT problem solved by EA and MFEA. The best value are highlighted in boldface.

Instances	EA			MFEA		
	BF	Avg	Avg	BF	Avg	Avg
10berlin52-2x5	1265672.9	1364770.7	1164646.4	1235853.7		
12eil51-3x4	70786.4	74726.4	66650.9	72092.9		
12eil76-3x4	183970.9	197678.8	159666.4	171642.9		
12pr76-3x4	40150030.5	42816754.4	35017437.6	37099269.8		
12st70-3x4	222752.5	235800.4	200185.7	211882.1		
15pr76-3x5	39342027.9	43671902.5	36084874.9	40125991.8		
16eil51-4x4	68868.2	73854.8	61079.9	67377.2		
16eil76-4x4	180451.0	195833.1	161420.6	170210		
16lin105-4x4	12404567.1	12957738.6	10755239.4	11622527.4		
16st70-4x4	217573.7	234338.3	198685.6	214505.5		
18pr76-3x6	41023837.1	43177318.2	34103588.2	353399402.2		
20eil51-4x5	72362.6	76595.1	65334.0	69527.5		
20eil76-4x5	178006.8	193912.2	158730.3	170962.8		
20st70-4x5	224071.9	240294.9	197197.6	218051.4		
25eil101-5x5	363706.0	404274.6	326204.4	353615.7		
25eil51-5x5	76000.5	80150.8	64662.7	70299.9		
25eil76-5x5	185727.3	201911.2	152927.8	176593.9		
25rat99-5x5	775778.4	842529.3	672730.4	734618.1		
28kroA100-4x7	16594234.3	18057077.9	13127755.6	15332145.5		
30kroB100-5x6	16197360.5	17777465.4	13321104.8	15180119.9		
35kroB100-5x5	15975455.4	17175668.5	13207171.0	14039628.2		

Small Instances

Large Instances

Instances	EA		MFEA		Avg	BF	Avg		
	BF	Avg	BF	Avg					
36eil101-6x6	400479.2	444799.2	301954.6	336621.4	10berlin52	1212367.7	1270287.3	1070892.2	1128537.5
42rat99-6x7	957728.7	1036331.1	726586.0	801111.5	10eil51	72400.6	75851	62220.9	68196.7
4berlin52-2x2	1204074.6	1389348.5	1407569.0	1468442.6	10eil76	182260.6	197520.7	154720.2	176318.9
4eil51-2x2	69289.7	79233.1	63352.3	74463.2	10kroB100	15544670.3	16793258.9	13410213.4	14430156.1
4eil76-2x2	193695.1	216040.9	160470.3	173950.9	10pr76	39803806.2	41978071.8	34443906.0	37528661.7
4pr76-2x2	43149835.2	48937295.2	37161565.6	39648471.3	10rat99	787308.7	844359.2	681410.1	733878.6
6berlin52-2x3	1250239.7	1387090.3	1211907.4	1338852.9	15eil51	71494.2	75430.9	65536.1	69262.8
6pr76-2x3	39811089.4	44505653.2	38034663.2	40323875.3	15eil76	171846.9	190699.4	152276.6	165567.5
6st70-2x3	244346.0	262236.4	210941.4	236206.3	15pr76	38723411.2	41355082	34406443.4	35902210.1
8berlin52-2x4	1335897.9	1412954.5	1161547.7	1248068.8	15st70	217633.3	229835	196100.4	213158.6
9eil101-3x3	382895.4	408022.9	323399.9	347184.1	25eil101	334091.7	382588.2	298566.2	331092.9
9eil51-3x3	73735.2	78254.5	65117.7	69709.3	25kroA100	15122586.0	16750515.8	13290236.0	14801709.7
9eil76-3x3	185572.5	200795.4	159175.3	171531	25lim105	10982060.6	11914969.5	155233.0	281917
18pr439-3x6	1416839349.6	1561793571.1	1025085049.5	1137123264.7	25rat99	784286.8	833241.1	638380.1	968377.2
20pr439-4x5	1477471558.3	1613475583	1305604456.7	1485293184.2	50eil101	479713.2	514738.4	9788.0	12654
25a280-5x5	12349778.6	13110863.9	10585336.1	11673178.5	50kroA100	21002827.6	22851798.3	22757582.7	30014740.7
25gil262-5x5	8614336.6	9381304.6	7589106.3	8232173.4	50kroB100	19047316.1	22503440.4	491024.2	606069.8
25pcb442-5x5	488506236.5	530402249.8	406965163.9	446099393.2	50lim105	15309391.6	17006431.4	14536077.1	23186711.8
36pcb442-6x6	508715616.5	553060407.9	354212969.5	408803078	50rat99	997133.3	1178858.6	1040326.9	1404530
42a280-6x7	12137547.4	14334181.3	9689989.9	10779328.7	5berlin52	1251233.9	1360740	27351.5	32335.8
49gil262-7x7	10277303.1	11437109.1	8263400.4	9904810.3	5eil51	73451.1	79603.6	1955.0	2198.5
49lin318-7x7	270088847.1	309400144.5	221116828.4	257834438.5	5eil76	198578.3	211390.5	129445.0	143326.5
9a280-3x3	14848459.2	15930228.1	12543766.2	14175273.4	5pr76	45587589.8	48401813.1	667837.8	822934.7
9gil262-3x3	10732649.9	11851918.2	9641118.4	10818917.3	5st70	240408.6	260969.9	168499.6	187038.6
9lin318-3x3	293171349.6	314380349	269984382.8	293792191.7	10a280	13907739.0	14844887.5	12300290.1	13350381.4
9pcb442-3x3	657120275.1	706429043.6	541118426.3	579184857.7	10gil262	10583888.3	11299878.5	9267931.7	10283586.5
					10lin318	274950582.8	297714185.9	237759185.5	270759068.6
					10pcb442	623457163.9	685302098.3	505781275.6	547646940.6
					10pr439	1390944006.2	1502442441.3	855937241.5	895455630.7
					25a280	10920850.4	11996040.1	9761001.7	10244430.5
					25gil262	8320804.1	9416167.1	7615763.2	8364538.5
					25lin318	225383682.6	251311451.3	195768659.3	211302877.7
					25pcb442	476613112.1	509512270.5	412614081.5	451957890.3
					25pr439	1162429357.3	1213987542.9	955864371.0	1022805770.1

Table 5: Details of best-found and average fitness per each instance of Type-1 CluMRC problem solved by EA and MFEA. The best value are highlighted in boldface.

Small Instances

Large Instances