Approximation Algorithms for the Balanced Optimization Splicing Problem in Undirected Graph

Yongsong Wen[1,2(✉)]

[1] School of Management, Wuhan University of Technology, Wuhan 430070, People's Republic of China
[2] School of Mathematics and Statistics, GuiZhou University of Finance and Economics, Guiyang 550025, People's Republic of China

Abstract. One-dimensional bin packing problem and balanced optimization problem are two classical problems in combinatorial optimization, inspired by this, we research a balanced optimization splicing problem: given a weight connected $G(V, E; w)$, a balanced spanning tree structure $\mathcal{A}$ and a specific material of fixed length ℓ, where $w : E \rightarrow Q^+$ (or $w : E \rightarrow Z^+$), each edge in the graph G is allowed to be greater than or equal to ℓ, we will use this specific material to splice a subgraph T from graph G with balanced spanning tree structure $\mathcal{A}$, and these edges spliced in such required structures are supposed to be cut from some pieces of a specific material of fixed length ℓ, the objective is to minimize the amount of special material used when splicing all the edges of the subgraph T with the given material. In this paper, we consider three kinds of balanced spanning tree structures for this problem and design three approximation algorithms.

Keywords: Balanced optimization · Balanced spanning tree · Approximation algorithms

1 Introduction and Problem Description

Consider the example of an area which wants to a signal stations network will be built to connect several towns, the construction period of the road linking any two towns is known, and all sections of the road commenced at the same time, we need design a signal station connected network to be finished at the same time as possible and meet the requirements to connect the signal station networks of cities and towns. In other words, the gap between maximum construction duration and minimum construction duration should be minimum. This example is a typical of the balanced optimization (BO) problem. Generally, the balanced optimization problem can be applied to the fair allocation of resources [1]. For further applications of BO problem and related problems we refer to [2] and [3]. We will give a brief introduction to the balanced optimization problem, let

H. Song and D. Jiang (Eds.): SIMUtools 2020, LNICST 369, pp. 263–273, 2021.
https://doi.org/10.1007/978-3-030-72792-5_24

$E = \{1, 2, \ldots, m\}$ be a finite set of m elements and F be a family of subsets of E, and $f : F \mapsto R$. Let w_e be a prescribed cost associated with each element e of E. For each $S \in F$, define $f(S) = \max_{e \in S} w_e - \min_{e \in S} w_e$. Then, the balanced optimization (BO)problem is to find $S \in F$ such that$f(S)$ is minimum. The balanced optimization problem was introduced by Martello et al., and they suggested a general algorithm, called the double threshold algorithm to solve the problem in polynomial time [3]. [4] considered balanced spanning tree problem and proposed an $\mathcal{O}(m^2)$ algorithm where m is the number of edges in the underlying graph. Later, Galil and Schieber [5] improved the algorithm presented in [4], the complexity of the Algorithm is $\mathcal{O}(mlogn)$, where n is the number of vertices considered in the graph. Wu [7] showed the balanced optimization problem with edge set restrictions(BOP-ESR), they provided efficient algorithms to solve the problem in polynomial time $\mathcal{O}(mn)$. Punnen and Nair [6] consider the balanced optimization problem with an additional linear constraint under a general combinatorial optimization setting. It is shown that this constrained balanced optimization (CBO) problem can be solved in polynomial time whenever an associated minimum problem can be solved in polynomial time. Other related special cases of BO problem can be referred to [8–12].

1.1 Problem Definition

In this paper, let's consider the previous example of BO problem again, in order to ensure connectivity different signal stations, we not only require that the between the maximum construction period and the minimum construction period is minimal, but we also need to use some special materials to connect the various signal stations, where the use of these special materials is limited in length, we objective is to use the least amount of special materials. We can regard each signal station as the vertex in the network and the route between stations as the path connected by the edge in the graph, so this problem can be regarded as a balanced optimization problem. In reality, when we splice such a balanced spanning subgraph with a special material, we generally need to consider the limitations of the length of the special material. Our motivation for considering the constraints is that each piece of specific material has a fixed length, and one piece of specific material can not splice all the edges of a balanced spanning tree structure $\mathcal{A}$ in this paper.

Furthermore, When we intend to splice an edge of any length $w(u, v)$ into such a balanced spanning tree structure, this splicing process can be roughly divided into two steps: (1) if one edge $(u, v) < \ell$, we need to "cut" a part of specific material, using a specific material of length $w(u, v)$ to splice this edge (u, v);(2) if an edge (u, v) $w(u, v) \geq \ell$, we first use $r(u, v)$ whole pieces of such a specific material of length ℓ, and then "cut" a part of such a specific material with the length$w(u, v) - r(u, v) \cdot L$, splicing the edge (u, v), where $r(u, v) = \left\lceil \frac{w(u,v)}{L} \right\rceil - 1$. Through the (1) and (2) splicing process, each edge (u, v) in this balanced spanning tree structure, the length of the remaining edges is less than ℓ, in addition to the other $r(u, v)$ as a whole for such a specific material in balanced spanning tree structure. Thus, we need to consider the minimum amount of material of length ℓ to be used in the splicing process. So we study a new problem of splicing some required balanced structures in undirected graph, that is formally defined as follows: given a weight connected $G(V, E; w)$, a balanced spanning tree structure $\mathcal{A}$ and a specific material of fixed length ℓ, where $w : E \rightarrow Q^+$ (or $w : E \rightarrow Z^+$), each edge in the graph G is allowed to be greater than or equal to ℓ, we will use this specific material to splice a subgraph T from graph G with balanced spanning tree structure $\mathcal{A}$, and these edges spliced in such required structures are supposed to be cut from some pieces of a specific material of fixed length ℓ, the objective is to minimize the amount of special material used when splicing all the edges of the subgraph T with the given material.

For the new objective mentioned above, we consider the following three different structures $\mathcal{A}$: (1)If the structure is a minimal balanced spanning tree, we may call it the a minimal balanced spanning tree splicing($MBSTS$, abbreviations) problem; (2)If the structure is a minimal balanced spanning tree with edge set restricted problem, we may call it the minimal balanced spanning tree splicing problem with edge set restricted ($MBSTS$-ESR, abbreviations); (3)When the structure is a constrained balanced tree problems, we may call it the constrained balanced trees splicing ($CBTS$, abbreviations)problems.

1.2 Main Results

In this paper, for problems of $MBSTS$, $MBSTS$-ESR and $CBTS$, there is no approximation algorithm to solve these problems, nor is there a special version of polynomial time to solve them. Therefore, we design three kinds of approximation algorithms to solve these three problems. The organization of this paper is as follows. In Sect. 2, We give a general description of terminology and lemmas. In Sect. 3, when this is a $MBSTS$ problem, we design a $\frac{3}{2}$-approximation algorithm, and an asymptotic polynomial time approximation scheme($PTAS$). In Sect. 4, when this is a $MBSTS$-ESR problem, we present a $\frac{3}{2}$-approximation algorithm. In Sect. 5, when this is a $CBTS$ problem, we present an asymptotic $\frac{3}{2}$-approximation algorithm.

2 Terminology and Lemmas

We will use the packing problem to solve the balanced optimization splicing problem, we need to introduce some notation and terminology can be found in reference [14–17]. For the sake of the clarity, we regard a particular material of length ℓ as the capacity of a bin ℓ. If we are given an edge about an edge in the undirected graph, we expressed this weight of edge (u, v) as "item" of size $w(u, v)$. We can assume that the length of the edge $w(u, v)$ is allowed to exceed the length of ℓ. For each edge (u, v) of length $w(u, v)$, we might first use the material length ℓ of $r(u, v)$ as a whole and then "cut" the part at the $w(u, v)$ length, from a piece of material length ℓ, to splice the edge (u, v) together, where $r(u, v) = \left\lceil \frac{w(u,v)}{L} \right\rceil - 1$ and $w'(u, v) = w(u, v) - r(u, v) \cdot L$. This process means that this item of size $w(u, v)$ is "packed" into a bin with capacity ℓ. Further, if we can use m pieces of a specific material to splice all edges needed in the balance spanning subgraph T, expressed this process as the one that it is sufficient to use m pieces of such a specific material with length ℓ to splice all edges needed in the balance spanning graph.

We use three notations, $G(V, E; w; \ell)$, $G(V, E; w; E_0; \ell)$ and $G(V, E; w; B; \ell)$ to respectively represent an instance of the $BSTS$ problem, the $MBSTS$-ESR problem, the $CBTS$ problem, where E_0 is a set of required edges of an instance of the $MBSTS$-ESR problem, and B is a positive integer constraint condition on $CBTS$ problem. The two graphs G and T may have the same structure, and this case that $w(e) = f(e)$ for every edge $e \in E$ may be considered viable. Other undefined notations and terminology can be found in papadimotriou and steiglitz (1998) [17], schrijver (2003) [18]. The lemma used in this paper is as follows.

Lemma 1 *[5]. The Galil and Schieber designed a polynomial time algorithm to solves the most uniform problem in $\mathcal{O}(mlogn)$ time, where n is the number of vertex in connected graph G, m is the amount of edges in connected graph G.*

Lemma 2 *[6]. The double threshold algorithm (DT-Algorithm) can correctly solved constrained balanced optimization problem in $\mathcal{O}(mf(m))$, $\mathcal{O}(f(m))$ is the complexity of solving $SUM(a, b)$, where $SUM(a, b) : min \sum_{e \in S} w(e)$, subject to $S \in F(a, b) = \{S \in F : e \in S, implies\ a \leq w(e) \leq b\}$.*

Lemma 3 *[7]. The Wu's improved algorithm correctly solves the minimum balanced spanning tree with edge restrictions and runs in $\mathcal{O}(mn)$ time.*

Lemma 4 *[16]. Suppose that m items $b_1, b_2, \ldots, b_m$ have sizes $w(b_1), w(b_2), \ldots, w(b_m)$, respectively, If we use the fist-fit-decreasing algorithm (FFD, abbreviations) to pack these mitems into some bins with capacity 1, the total size of each bin generated by the algorithm $FFD$is larger than $\frac{2}{3}$, with the exception of the last bin.*

Lemma 5 *[16]. Suppose that m items $b_1, b_2, \ldots, b_m$ have sizes $w(b_1), w(b_2), \ldots, w(b_m)$, respectively, where $0 < w(b_i) \leq \frac{1}{2}$. When we use the fist-fit-decreasing algorithm (FFD, for short) to pack these m items into some bins with capacity 1, then the total size of items packed into each bin produced by the algorithm FFD is greater than $\frac{2}{3}$, except the last bin used.*

Lemma 6 *[18]. For any edge-weighted graph $G(V, E; w)$, $w : E \to Q^+$ is a weight function. If a minimum spanning tree T of G has the edge set $E_T = \{e_{i_1}, e_{i_2}, \ldots, e_{i_{n-1}}\}$ to satisfy: $w(e_{i_1}) \leq w(e_{i_2}) \leq \cdots \leq w(e_{i_{n-1}})$, and if any spanning tree T' of G has the edge set $E_T = \{e_{j_1}, e_{j_2}, \ldots, e_{j_{n-1}}\}$ to satisfy: $w(e_{j_1}) \leq w(e_{j_2}) \leq \cdots \leq w(e_{j_{n-1}})$, then $w(e_{i_k}) \leq w(e_{j_1}), k = 1, 2, \ldots, n-1$.*

3 The Minimal Balanced Spanning Tree Splicing Problem

In this section, in the case of $MBSTS$ problem, given a balanced spanning tree $T = (V, E; w)$ as an instance, it can become the bin packing problem with the n items $w(e)$ as an input. In other word, the bin packing problem polynomial time can reduction to splice balanced spanning tree problem, due to bin packing problem is NP-hard, so the splicing balanced spanning tree problem is also NP-hard [15]. The approximate algorithm of $MBSTS$ problem is described as follows.

Algorithm 3.1: $MBSTS$

Input :	A simple connected weighted graph $G(V, E; w; \ell)$.
Output:	A balanced spanning tree $T = G(V, E')$ and the number of bins used.
Step 1	A minimum balanced spanning tree $T = G(V, E')$ in simple connected weighted graph $G(V, E; w; \ell)$ is obtained by using the balanced spanning tree algorithm [5], where E'=$\{e_{i_1}, e_{i_2}, \dots, e_{i_{n-1}}\}$;
Step 2	For $\forall e \in T$, let $insert(e) = \lceil \frac{w(e)}{\ell} \rceil - 1$, then we use $insert(e)$ pieces of whole material to splice part of edge e, and denote $m_0 = \sum_{e \in T} insert(e)$ as the number of special materials used in this step;
Step 3	For $\forall e \in T$, let $w'(e) = w(e) - insert(e) \cdot \ell$, $w'(e_{i_1}), w'(e_{i_1}), \dots, w'(e_{i_{n-1}})$ are the length of remaining edge e in the balanced spanning tree T that are spliced with the whole special materials. We regard the weight of these edges of $w'(e_{i_1}), w'(e_{i_1}), \dots, w'(e_{i_{n-1}})$ as the size of the items, then we use the FFD algorithm to pack $w'(e_{i_1})$,$w'(e_{i_1}), \dots, w'(e_{i_{n-1}})$ into bins with capacity ℓ, and denote m_1 as the number of bins used this step;
Step 4	Output a balanced spanning tree T and the number of $m_0 + m_1$ bins used in this algorithm, i.e. $OUT = m_0 + m_1$.

Using the 3.1 algorithm, we get the result of the $BSTS$ problem as follows:

Theorem 1. *Algorithm 3.1 is a $\frac{3}{2}$-approximation algorithm to solve the $BSTS$ problem, that is, the algorithm satisfies $OUT \le \frac{3}{2} \cdot OPT$, where OUT refers to the numbers of special materials output by the algorithm 3.1, OPT refers to the numbers of optimization materials required for the $BSTS$ problem, and its running time is $\mathcal{O}(nlogn)$.*

Proof. Suppose that $T = G(V, E')$ is a minimum balanced spanning tree produced by algorithm 3.1 with the value OUT, we assume that $T = G(V, E')$ has the edges set E'=$\{e_{i_1}, e_{i_2}, \dots, e_{i_{n-1}}\}$, and satisfying $w(e_{i_1}) \le w(e_{i_2}) \le \cdots \le w(e_{i_{n-1}})$, where $n = |V|$. We also assume that $T^* = G(V, E^*)$ is an optimal balanced spanning tree to the instance G for the $BSTS$ problem with the value OPT, and the corresponding edge set satisfies $w(e_{j_1}) \le w(e_{j_2}) \le \cdots \le w(e_{j_{n-1}})$. By lemma 2.6, we can obtain this inequality $w(e_{i_k}) \le w(e_{j_k})$ holds for each $k = 1, 2, \dots, n-1$.

Now let's think about the splicing scheme for tree T, follow the previous analysis in Sect. 2, we can divide the splicing of tree T into two stages for analysis:(1) let $insert(e) = \lceil \frac{w(e)}{\ell} \rceil - 1$, let's first cover the length of edges e with the whole special materials ℓ for all edges of length greater than or equal to ℓ in T, until the length of all edges is less than ℓ. (2) let $w'(e) = w(e) - insert(e) \cdot \ell$, we use the FFD algorithm to pack all the remaining edges of the tree T with length $w'(e_{i_k})(k = 1, 2, \ldots, n-1)$ into bins, and denote m_1 as the number of bins used this step, but the theoretical optimal value is m^* in this step, so there is $m_1 \leq \frac{3}{2} \cdot m^*$. Then we obtain the fact $OUT = \sum_{e_{i_k} \in T} insert(e_{i_k}) + m_1 \leq \sum_{e_{i_k} \in T} insert(e_{i_k}) + \frac{3}{2} m^* \leq \frac{3}{2}(\sum_{e_{i_k} \in T} insert(e_{i_k}) + m^*) \leq \frac{3}{2} \cdot OPT_T$.

Since $T^* = G(V, E^*)$ is an optimal balanced spanning tree to the instance G for the $CBTS$ problem, and the items $w(e_{i_1}), w(e_{i_2}), \ldots, w(e_{i_{n-1}})$ are packed into OPT bins with capacity ℓ, we also can pack these items $w(e_{i_1}), w(e_{i_2}), \ldots,$ $w(e_{i_{n-1}})$ of T into these OPT bins as follows:(1) if $insert(e_{i_k}) = insert(e_{j_k})$, we use the optimal packing method of T^* to find $insert(e_{j_k})$ bins that can be pack into the bin with $insert(e_{i_k})$ edges in T, and these edges of length $w(e_{i_k}) - insert(e_{i_k}) \cdot \ell$ can be packed into $w(e_{j_k}) - insert(e_{j_k}) \cdot \ell$ bins. (2) if $insert(e_{i_k}) \leq insert(e_{j_k})$, we can pack this part of $w(e_{i_k}) - insert(e_{i_k}) \cdot \ell$ into one of $insert(e_{j_k}) - insert(e_{i_k})$ bins. It implies that the minimum number OPT_T of bins for the items $w(e_{i_1}), w(e_{i_2}), \ldots, w(e_{i_{n-1}})$ of T is no more than OPT, so we have the inequality $OPT_T \leq OPT$. Thus, we finally obtain $OUT \leq \frac{3}{2} \cdot OPT$ by the fact $OPT_T \leq OPT$.

Runtime analysis of Algorithm 3.1: by Lemma 1, step 1 needs running time $\mathcal{O}(mlogn)$ to find a minimal balanced spanning tree $T = G(V, E')$; at step 2, it's most running time $\mathcal{O}(n)$ pack all items into bins; at step 3, FFD algorithm needs at most running time $\mathcal{O}(nlogn)$ to pack $n-1$ items of lengths of T into bins with length ℓ. Hence, the whole algorithm needs the running time $\mathcal{O}(nlogn)$.

In step 4 of Algorithm 3.1, when we use the bin packing algorithm to solve the $MBSTS$ problem, we use an asymptotic polynomial time approximation scheme $(PTAS)\mathcal{A}_\varepsilon$ [15] instead of the FFD algorithm, that is, using an asymptotic $PTAS$ $\mathcal{A}_\varepsilon$ [15] to pack $w'(e_{i_1})$,$w'(e_{i_1}), \ldots,$ $w'(e_{i_{n-1}})$ into m_1 bins with capacity ℓ in the step 3. Thus, we can get a improved algorithm for the $MBSTS$ problem, the conclusion is as follows.

Theorem 2. *For any* $0 < \varepsilon \leq \frac{1}{2}$ *and given an instance* $G(V, E; w; \ell)$ *of the* $MBSTS$ *problem, there is an algorithm* $\mathcal{A}_\varepsilon$ *that runs in time polynomial in* n*, and the improved algorithm 3.1 can get at most*$(1+2\varepsilon)OPT+1$ *special materials of length* ℓ *to splice all edges in a balanced spanning tree* T*, where the improved algorithm 3.1 is an asymptotic* $PTAS$ *to solve the* $MBSTS$ *problem.*

Proof. In the third step of algorithm 3.1, use an asymptotic $PTAS$ $\mathcal{A}_\varepsilon$ [15] to pack $w'(e_{i_1})$,$w'(e_{i_1}), \ldots,$ $w'(e_{i_{n-1}})$ into m_1 bins with capacity ℓ in a balanced spanning tree, and an asymptotic $PTAS$ $\mathcal{A}_\varepsilon$ can obtain the output value of a feasible solution m_1 to satisfy $m_1 \leq (1 + 2\varepsilon)(OPT_T - m_0) + 1$. The same proof is used for algorithm 3.1, we also have $OPT_T \leq OPT$. From what has been discussed above, for each value of ε, we get $OUT = m_0 + m_1 \leq m_0 + (1 + 2\varepsilon)(OPT_T - m_0) + 1 \leq (1 + 2\varepsilon)OPT - 2\varepsilon \cdot m_0 + 1 \leq (1 + 2\varepsilon)OPT + 1$.

4 The Balanced Spanning Tree Splicing Problem with Edge Set Restricted

In this section, we consider the splicing balanced spanning tree problem with edge set restricted ($MBSTS$-ESR). The argument for the $MBSTS$-ESRproblem is similar to that for the $MBSTS$problem, the $MBSTS$-ESR problem also can not be approximated within performance ration $\frac{3}{2} - \varepsilon$, $\forall \varepsilon > 0$, unless $P = NP$. We now can present an approximation algorithm to solve the $MBSTS$-ESR problem as follows.

Algorithm 4.1: $MBSTS$-ESR

Input:	A simple connected weighted graph $G(V, E; w; E_0; \ell)$
Output:	A minimum balanced spanning tree $T = G(V, E')$ with edge set Restricted, and the number of m bins used.
step 1	A minimum balanced spanning tree $T = G(V, E')$ with edge set Restricted in graph $G(V, E; w; E_0; \ell)$ is obtained by using the balanced spanning tree algorithm with edge set restricted [7], where $E_0 \subset E'=\{e_{i_1}, e_{i_2}, \ldots, e_{i_{n-1}}\}$;
step 2	For $\forall e \in T$, let $insert(e) = \lceil \frac{w(e)}{\ell} \rceil - 1$, use complete special materials ℓ to splice part of edge e, and denote $m_0 = \sum_{e \in T} insert(e)$;
step 3	For $\forall e \in T$, let $w'(e) = w(e) - insert(e) \cdot \ell$, we denote the weight of these edges by $w'(e_{i_1}), w'(e_{i_1}), \ldots,$ $w'(e_{i_{n-1}})$ as the size of the items, then we use the FFD algorithm to pack $w'(e_{i_1})$,$w'(e_{i_1}), \ldots,$ $w'(e_{i_{n-1}})$ into bins with capacity ℓ, and denote m_1 as the number of bins used this step;
step 4	Output a balanced spanning tree T and the number of $OUT = m_0 + m_1$ bins used in this algorithm.

Using the Algorithm 4.1, we obtain the following result for the $MBSTS$-ESR problem.

Theorem 3. *Algorithm 4.1 is a $\frac{3}{2}$-approximation algorithm to solve the $MBSTS$-ESR problem, that is the algorithm satisfies $OUT \leq \frac{3}{2} \cdot OPT$, where OUT refers to the numbers of special materials output by the algorithm, OPT refers to the numbers of optimization materials required for the $MBSTS$-ESR problem, and its running time is $\mathcal{O}(mn)$.*

Proof. Suppose that $T = G(V, E')$ is a minimum balanced spanning tree with edge set restricted produced by algorithm 4.1, having the output value $OUT = m_0 + m_1$, we assume that $T = G(V, E')$ has the edges set $E' = \{e_{i_1}, e_{i_2}, \ldots, e_{i_{n-1}}\}$, and satisfying $w(e_{i_1}) \leq w(e_{i_2}) \leq \cdots \leq w(e_{i_{n-1}})$, where $n = |V|$,$E_0 \subset E'$. We also assume that $T^* = G(V, E^*)$ is an optimal balanced spanning tree to the instance G for the $BSTS$ problem with the value OPT, and the corresponding edge set satisfies $w(e_{j_1}) \leq w(e_{j_2}) \leq \cdots \leq w(e_{j_{n-1}})$, $E_0 \subset E^*$. By Lemma 6, we can obtain this inequality $w(e_{i_k}) \leq w(e_{j_k})$ holds for each $k = 1, 2, \ldots, n - 1$.

Now let's think about the splicing scheme for tree T with edge set restricted, the proof method of this algorithm is similar to Theorem 1, this splice process can be broadly broken into two steps:(1) let $insert(e) = \lceil \frac{w(e)}{\ell} \rceil - 1$, let's cover the length of edges e with the whole special materials ℓ for all edges of length greater than or equal to ℓ in T, until the length of all edges is less than ℓ. (2) let $w'(e) = w(e) - insert(e) \cdot \ell$, we use the FFD algorithm to pack all the remaining edges of the tree T with length $w'(e_{i_k})(k = 1, 2, \dots, n-1)$ into bins, and denote m_1 as the number of bins used this step, but the theoretical optimal value is m^* in this step, so there is $m_1 \leq \frac{3}{2} \cdot m^*$. Then we obtain the fact $OUT = \sum_{e_{i_k} \in T} insert(e_{i_k}) + m_1 \leq \sum_{e_{i_k} \in T} insert(e_{i_k}) + \frac{3}{2} m^* \leq \frac{3}{2}(\sum_{e_{i_k} \in T} insert(e_{i_k}) + m^*) \leq \frac{3}{2} \cdot OPT_T$.

Due to $T^* = G(V, E^*)$ is an optimal balanced spanning tree with edge set restricted to the instance G of the $MBSTS$-ESR problem, and the items $w(e_{i_1}), w(e_{i_2}), \dots, w(e_{i_{n-1}})$ are packed into OPT bins with capacity ℓ, we also can pack these items $w(e_{i_1}), w(e_{i_2}), \dots,$ $w(e_{i_{n-1}})$ of T into these OPT bins as follows:(1) if $insert(e_{i_k}) = insert(e_{j_k})$, we use the optimal packing method of T^* to find $insert(e_{j_k})$ bins that can be pack into the bin with $insert(e_{i_k})$ edges in T, and these edges of length $w(e_{i_k}) - insert(e_{i_k}) \cdot \ell$ can be packed into $w(e_{j_k}) - insert(e_{j_k}) \cdot \ell$ bins. (2) if $insert(e_{i_k}) \leq insert(e_{j_k})$, we can pack this part of $w(e_{i_k}) - insert(e_{i_k}) \cdot \ell$ into one of $insert(e_{j_k}) - insert(e_{i_k})$ bins. It show that the minimum number OPT_T of bins for the items $w(e_{i_1}), w(e_{i_2}), \dots, w(e_{i_{n-1}})$ of T is no more than OPT, so we have the inequality $OPT_T \leq OPT$. Thus, we finally obtain $OUT \leq \frac{3}{2} \cdot OPT$ by the fact $OPT_T \leq OPT$.

Runtime analysis of Algorithm 4.1: in step 1 of Algorithm 4.1, by Lemma 3, we know that computing the running time of a balanced spanning tree $T = G(V, E')$ with edge set restricted is $\mathcal{O}(mn)$; in step 2 of algorithm 4.1, it's running time is $\mathcal{O}(n)$ is $\mathcal{O}(mn)$; in step 3 of Algorithm 4.1, it's most running time is $\mathcal{O}(nlogn)$. Hence, the algorithm 4.1 needs the running time $\mathcal{O}(mn)$.

This completes the proof of the Theorem 3 mentioned-above.

5 The Constrained Balanced Spanning Tree Splicing Problem

In this section, we consider the splicing problem of constrained balanced spanning tree($CBTS$), use a simple connected graph $G(V, E; w; B; \ell)$ as instance of the splicing constrained balanced spanning tree problem, where $w : E \rightarrow Z^+$ is a non-negative function, and B is positive integer. There are some special materials with length ℓ, we need find a balanced spanning tree $T \subset F$ from the undirected graph G, it satisfies $w(T) = \sum_{e \in T} w(e) \leq B$, so that each edge in the balanced spanning graph T is spliced by a part of a piece (or some whole pieces)of a special material of length ℓ, where F be a family of subsets of E. The objective is to minimize the number of necessary pieces of such a specific material to splice all edges needed in constrained balanced spanning tree T.

For the instance of $G(V, E; w; B; \ell)$, if we can find a balanced spanning tree $T \subset F$ from the graph G, it satisfies $w(T) = \sum_{e \in T} w(e) \leq B$, and the length of all edges e in the tree T does no longer than the length of the special material ℓ, then the $CBTS$ problem is the promotion form of the packing problem. According to the inapproximability of the packing problem, the $CBTS$ problem can not be approximated within performance ration $\frac{3}{2} - \varepsilon$, $\forall \varepsilon > 0$, unless $P = NP$. We need design an asymptotic approximation algorithm to solve the $CBTS$ problem, and the method of the algorithm is as follows:(1)for each edge e in graph G, we denote $insert(e) = \lceil \frac{w(e)}{\ell} \rceil - 1$, $w'(e) = insert(e) \cdot \ell$, $w''(e) = w(e) - w'(e)$, and if $\frac{\ell}{2} < w''(e) < \frac{2\ell}{3}$, then $\theta(e) = 1$, otherwise,$\theta(e) = 0$; (2) given a simple connected weighted graph $G(V, E; w; \ell; B)$, we used the double threshold algorithm (DT-algorithm) [6] to find a constrained minimum balanced spanning tree T in graph G. (3) use the FFD algorithm [16] for the bin packing problem to pack the items of lengths of edges in T into some bins with capacity ℓ.

Algorithm 5.1: $CBTS$

Input:	a simple connected weighted graph $G(V, E; w; B; \ell)$
Output:	the number of bins used, and a minimum balanced spanning tree $T = G(V, E')$, where $T \subset F$, $w(T) = \sum_{e \in T} w(e) \leq B$.
Step 1	Use the DT-algorithm to find a minimum balanced spanning tree $T = G(V, E')$ in $G(V, E; w; \ell; B)$, if $w(T) = \sum_{e \in T} w(e) > B$, then output "There is no feasible solution for this instance", the algorithm stopped; if $w(T) = \sum_{e \in T} w(e) \leq B$ and $T \subset F$, where $E'(T)=\{e_{i_1}, e_{i_2}, \ldots, e_{i_{n-1}}\}$, then go to the next step;
Step 2	For each edge e in graph G, we denote $insert(e) = \lceil \frac{w(e)}{\ell} \rceil - 1$, $w'(e) = insert(e) \cdot \ell$, $w''(e) = w(e) - w'(e)$, and if $\frac{\ell}{2} < w''(e) < \frac{2\ell}{3}$, then $\theta(e) = 1$, otherwise, $\theta(e) = 0$;
Step 3	For $\forall e \in E'$, the part of size $w'(e)$ on edge e is packed into $insert(e)$ bins with capacity ℓ, and denotes $m_0 = \sum_{e \in E'} insert(e)$ as the number of bins used in this step.
Step 4	For items of size $w''(e_{i_1}), w''(e_{i_1}), \ldots, w''(e_{i_l})$, use the FFD algorithm to pack $w''(e_{i_1})$, $w''(e_{i_1})$, $\ldots$, $w''(e_{i_{n-1}})$ into bins with capacity ℓ, and denote m_1 as the number of bins used this step;
Step 5	Output a minimum balanced spanning tree T and the number of $m_0 + m_1$ bins used in this algorithm, i.e. $OUT = m_0 + m_1$.

By the Algorithm 5.1, we obtain the following result for the $CBTS$ problem.

Theorem 4. *Algorithm 5.1 is an asymptotic $\frac{3}{2}$-approximation algorithm to solve the $CBTS$ problem, that is the algorithm satisfies $OUT \leq \frac{3}{2} \cdot OPT + \frac{1+r_0}{4}$, where OUT refers to the numbers of special materials output by the algorithm, OPT refers to the numbers of optimization materials required for $CBTS$ problem, r_0 denote the number of edges on the tree T where the remaining length satisfies $\frac{\ell}{2} < w''(e) < \frac{2\ell}{3}$, and its running time is $\mathcal{O}(mf(m))$.*

Proof. Suppose that $T^* = G(V, E^*)$ is an optimal balanced spanning tree to the instance G for the $CBTS$ problem with the value OPT, where $T^* \subset F$, $w(T^*) = \sum_{e \in T} w(e) \leq B$. We also assume that $T = G(V, E')$ is a minimum balanced spanning tree produced by Algorithm 5.1 with the value $OUT = m_0 + m_1$.

Since T is a minimum balanced spanning tree for weight function $w(\cdot)$ in simple connected weighted undirected graph G, we get $\sum_{e\in T} w(e) \le \sum_{e\in T^*} w(e) \le \ell\cdot OPT$ by Lemma 6. For every edge e in a balanced spanning tree T, we first use $insert(e)$ pieces of whole special materials of length ℓ to splice them, and then the remaining length of the unspliced part of edge e in T is denoted as $w''(e)$, it indicates that m_0 pieces of whole special materials of length ℓ were used in the step 3. For items of size $w''(e_{i_1}), w''(e_{i_1}), \ldots, w''(e_{i_l})$ in the step 4, we use the following notation, let r_0 denote the number of remaining edges in the tree T that satisfy $\frac{\ell}{2} < w''(e) < \frac{2\ell}{3}$, use r_1 denote the number of remaining edges in the tree T should be $\frac{2\ell}{3} \le w''(e)$. The m_1 bins used in the step 4 are listed as $B_1, B_2, \ldots, B_{m_1}$, respectively, and the sum of the items in each bin B_i is denoted as $f(B_i)(\ i = 1, 2, \ldots, m_1)$, so we can get the following facts.

(1) $f(B_i) \ge \frac{2\ell}{3}$, $i = 1, 2, \ldots, r_1$;
(2) $f(B_i) \ge \frac{\ell}{2}$, $i = 1 + r_1, 2 + r_1, \ldots, r_1 + r_0$; (by Lemma 4)
(3) $f(B_i) > \frac{2\ell}{3}$, $i = r_1 + r_0 + 1, r_1 + r_0 + 2, \ldots, m_1 - 1$; (by Lemma 5)
(4) $f(B_i) + f(B_{m_1}) > \ell$, $i = 1, 2, \ldots, m_1 - 1$.

Thus, we can get

$w(T) = \sum_{e\in T} w(e) = \sum_{e\in T} w'(e) + \sum_{e\in T} w''(e) = \sum_{e\in T} insert(e)\cdot\ell + \sum_{i=1}^{m_1} f(B_i) = m_0\cdot\ell + \sum_{i=1}^{r_1} f(B_i) + \sum_{i=r_1+1}^{r_1+r_0} f(B_i) + \sum_{i=r_1+r_0+1}^{m_1-1} f(B_i) + f(B_{m_1}) = m_0\cdot\ell + \sum_{i=1}^{r_1} f(B_i) + \sum_{i=r_1+1}^{r_1+r_0}(f(B_i) + \frac{\ell}{6}) + \sum_{i=r_1+r_0+1}^{m_1-1} f(B_i) + f(B_{m_1}) - \frac{r_0}{6}\cdot\ell > m_0\cdot\ell + \frac{2\ell}{3}(m_1 - 2) + \ell + \frac{r_0}{6}\cdot\ell \ge \frac{2\ell}{3}\cdot(m_0 + m_1) - \frac{r_0+2}{6}\cdot\ell = \frac{2\ell}{3}\cdot OUT - \frac{r_0+2}{6}\cdot\ell$.

In summary, we know $\frac{2\ell}{3}\cdot OUT - \frac{r_0+2}{6}\cdot\ell < \sum_{e\in T} w(e) \le \ell\cdot OPT$, implying $4OUT < 6OPT + r_0 + 2$. Hence, we obtain the fact $OUT \le \frac{3}{2}OPT + \frac{r_0+1}{4}$ by the integral property.

Runtime analysis of Algorithm 5.1: at step 1, by Lemma 2, the running time of the DT-algorithm is $\mathcal{O}(mf(m))$; at step 3, it's needs most running time is $\mathcal{O}(n)$; at step 4, the FFD algorithm needs at most running time $\mathcal{O}(nlogn)$. Hence, the whole algorithm needs the running time $\mathcal{O}(mf(m))$. The running time of the FFDalgorithm is $\mathcal{O}(mf(m))$. Hence, the whole algorithm needs the running time $\mathcal{O}(mf(m))$.

6 Conclusions

In this paper, we proposed three kinds of algorithms, and obtain three main results:(1) for the $MBSTS$ problem, we obtain an $\frac{3}{2}$-approximation algorithm and an asymptotic $PTAS$, algorithm 3.1 running time is $\mathcal{O}(nlogn)$; (2) for the $MBSTS$-ESR problem, we obtain an $\frac{3}{2}$-approximation algorithm, and algorithm 4.1 has its running time $\mathcal{O}(nm)$; (3)for the $CBTS$ problem, we get an asymptotic $\frac{3}{2}$-approximation algorithm, and Algorithm 5.1 has its running time $\mathcal{O}(mf(m))$. Further improvement requires deeper exploration. The most hardest work is to design approximation algorithms with lower constant approximation ratios to solve the $CBTS$(or$MBSTS$-ESR) problem.

References

1. Punnen, A.P., Taghipour, S., Karapetyan, D., Bhattacharyya, B.: The quadratic balanced optimization problem. Discret. Optim. **12**(2), 47–60 (2014). https://doi.org/10.1016/j.disopt.2014.01.001
2. Duin, C.W., Volgenant, A.: Minimum deviation and balanced optimization: a unifed approach. Oper. Res. Lett. **10**(1), 43–48 (1991). https://doi.org/10.1016/0167-6377(91)90085-4
3. Martello, S., Pulleyblank, W.R., Toth, P., Werra, D.: Balanced optimization problems. Oper. Res. Lett. **3**(5), 275–278 (1984). https://doi.org/10.1016/0167-6377(84)90061-0
4. Camerni, P.M., Maffioli, F., Martello, S., Toth, P.: Most and least uniform spanning trees. Discret. Appl. Math. **15**(2), 181–197 (1986). https://doi.org/10.1016/0166-218X(86)90041-7
5. Galil, Z., Schieber, B.: On finding most uniform spanning trees. Discret. Appl. Math. **20**(2), 173–175 (1988). https://doi.org/10.1016/0166-218X(88)90062-5
6. Punnen, A.P., Nair, K.P.K.: Constrained balanced optimization problems. Comput. Math. Appl. **37**, 157–163 (1999). https://doi.org/10.1016/S0898-1221(99)00119-4
7. Wu, L.: An efficient algorithm for the most balanced spanning tree problems. Adv. Sci. Lett. **11**, 776–778 (2012). https://doi.org/10.1166/asl.2012.3033
8. Punnen, A.P., Aneja, Y.P.: Lexicographic balanced optimization problems. Oper. Res. Lett. **32**(1), 27–30 (2004). https://doi.org/10.1016/S0167-6377(03)00065-8
9. Gupta, S.K., Punnen, A.P.: Minimum deviation problems. Oper. Res. Lett. **7**(4), 201–204 (1988). https://doi.org/10.1016/0167-6377(88)90029-6
10. Cappanera, P., Scutella, M.G.: Balanced paths in acyclic networks: Tractable cases and related approaches. Networks **45**(2), 104–111 (2005). https://doi.org/10.1002/net.20053
11. Ahuja, R.K.: The balanced linear programming problem. Eur. J. Oper. Res. **101**(1), 29–38 (1997). https://doi.org/10.1016/S0377-2217(96)00142-7
12. Eppstein, D.: Minimum range balanced cuts via dynamic subset sums. J. Algorithms **23**(2), 375–385 (1997). https://doi.org/10.1006/jagm.1996.0841
13. Li, J., Ge, Y., He, S., et al.: Approximation algorithms for constructing some required structures in digraphs. Eur. J. Oper. Res. **232**(2), 307–314 (2014). https://doi.org/10.1016/j.ejor.2013.07.033
14. LaRusic, J., Punnen, A.P.: The balanced traveling salesman problem. Comput. Oper. Res. **38**(5), 868–875 (2011)
15. Vazirani, Vijay V.: Approximation Algorithms. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-662-04565-7
16. Simchi-Levi, D.: New worst case results for the bin-packing problem. Nav. Res. Logist. **41**(4), 579–858 (1994). https://doi.org/10.1002/1520-6750(199406)41:4
17. Papadimitiou, C.H., Steiglitz, K.: Combinatorial Optimization: Algorithms and Complexity. Dover, New York (1988). https://doi.org/10.1109/TASSP.1984.1164450
18. Schrijver, A.: Combinatorial Optimization: Poluhedra and Efficiency. Springer, Dordrecht (2003)