ELSEVIER Discrete Applied Mathematics 49 (1994) 25–50

DISCRETE APPLIED MATHEMATICS

Approximation algorithms for multi-dimensional assignment problems with decomposable costs

Hans-Jürgen Bandelt[a], Yves Crama[b], Frits C.R. Spieksma[c]

[a]*Mathematisches Seminar der Universität Hamburg, Bundesstr. 55, D-2000 Hamburg 13, Germany*
[b]*Department of Quantitative Economics, University of Limburg, P.O. Box 616, 6200 MD Maastricht, Netherlands*
[c]*Department of Mathematics, University of Limburg, P.O. Box 616, 6200 MD Maastricht, Netherlands*

Received 27 June 1991; revised 17 February 1992

Abstract

The k-dimensional assignment problem with decomposable costs is formulated as follows. Given is a complete k-partite graph $G = (X_0 \cup \cdots \cup X_{k-1}, E)$, with $|X_i| = p$ for each i, and a nonnegative length function defined on the edges of G. A clique of G is a subset of vertices meeting each X_i in exactly one vertex. The cost of a clique is a function of the lengths of the edges induced by the clique. Four specific cost functions are considered in this paper; namely, the cost of a clique is either the sum of the lengths of the edges induced by the clique (sum costs), or the minimum length of a spanning star (star costs) or of a traveling salesman tour (tour costs) or of a spanning tree (tree costs) of the induced subgraph. The problem is to find a minimum-cost partition of the vertex set of G into cliques. We propose several simple heuristics for this problem, and we derive worst-case bounds on the ratio between the cost of the solutions produced by these heuristics and the cost of an optimal solution. The worst-case bounds are stated in terms of two parameters, viz. k and τ, where the parameter τ indicates how close the edge length function comes to satisfying the triangle inequality.

Key words: Multi-dimensional assignment; Triangle inequality; Heuristics; Worst-case performance

1. Introduction

For $k \geqslant 2$, the k-dimensional assignment problem is formulated as follows. Let $X_0, \ldots, X_{k-1}$ be pairwise disjoint sets of equal cardinality, say p. Regarding $V = \bigcup_i X_i$ as the vertex set of a complete k-partite graph with edge set $E = \bigcup_{i<j} \{\{u, v\} \mid u \in X_i, v \in X_j\}$, we say that a subset X of V is a *clique* of the graph (V, E) if X meets every X_i in exactly one vertex ($i = 0, \ldots, k-1$). A (*k-dimensional*) *assignment of* (V, E) is a partition of V into cliques, that is, a collection of p pairwise disjoint cliques of (V, E); we will

SSDI 0166-218X(92)00129-O

also call this an assignment between $X_0, X_1, \ldots,$ and X_{k-1}. Let now c be any real-valued cost function defined on the set of cliques of (V, E). The *k-dimensional assignment problem* on (V, E) with respect to c consists in finding an assignment M of minimum cost, where the cost of M is defined as $c(M) = \sum_{X \in M} c(X)$.

When $k = 2$, the k-dimensional assignment problem is nothing but the well-known bipartite weighted matching problem and can be solved in $O(p^3)$ arithmetic operations by the Hungarian method (see e.g. Papadimitriou and Steiglitz [11]). Throughout this paper, we refer to 2-dimensional assignments as to *matchings*.

The 3-dimensional assignment problem has also been actively investigated in the literature; see e.g. the references contained in Balas and Saltzman [1], as well as Crama and Spieksma [3], Frieze [5], Hansen and Kaufman [8]; it is well known to be NP-hard [9]. The k-dimensional assignment problem has been less thoroughly studied for values of $k \geqslant 4$, and this, in spite of the fact that it constitutes a most natural generalization of the 2- and 3-dimensional cases. Early mention of the problem can be found in Haley [7] and in Pierskalla [12], where applications are described.

In this paper, we concentrate on the restricted version of the k-dimensional assignment problem which arises when the cost of a clique is not completely arbitrary, but is rather a function of elementary costs attached to the edges of the complete k-partite graph (V, E). Specifically, let us assume from now on that d is a nonnegative length function defined on E, and not identically zero; for the sake of simplicity, we use the shorthand $d(u, v)$ instead of $d(\{u, v\})$, and we let $d(v, v) = 0$, by convention. We say that the cost function c is *decomposable* if there exists a function $f: R^{\binom{k}{2}} \to R$ such that, for every clique $X = \{x_0, \ldots, x_{k-1}\}$,

$$c(X) = f(d(x_0, x_1), d(x_0, x_2), \ldots, d(x_{k-2}, x_{k-1})).$$

Thus, $c(X)$ is completely determined by the lengths of the edges induced by X.

Different variants of the 3-dimensional assignment problem with decomposable costs arise in applications considered by Frieze and Yadegar [6] or Crama et al. [2], and are further investigated in Crama and Spieksma [3].

The main goal of this paper is to present some simple heuristics for the k-dimensional assignment problem with decomposable costs and, for various cost functions, to derive worst-case bounds on the ratio between the cost of the heuristic assignments and the cost of an optimal solution.

In the next section, we describe the specific decomposable cost functions which will be considered in the remainder of the paper. In Section 3, we propose some heuristics for the k-dimensional assignment problem with decomposable costs, and we state our main results about the worst-case performance of these heuristics. Proofs of these results are to be found in Sections 4–8 (Sections 4–7 deal with a first type of heuristics and four different cost functions, while Section 8 focusses on a second type of heuristics).

The worst-case bounds are stated in terms of two parameters, viz. k and τ, where τ is the smallest real number for which the following condition holds: for all $\{u, v\}$, $\{u, w\}$, $\{v, w\} \in E$,

$$d(u, v) \leqslant \tau(d(u, w) + d(v, w)). \tag{1.1}$$

Observe that τ is well defined, except when there exist three edges $\{u,v\}$, $\{u,w\}$, $\{v,w\}$ such that $d(u,v) > 0$ and $d(u,w) = d(v,w) = 0$. In the latter case, we let $\tau = \infty$.

Clearly, $\tau \geqslant \frac{1}{2}$, and $\tau = \frac{1}{2}$ exactly when d is constant on E. If $\tau \leqslant 1$, then the edge lengths satisfy the usual triangle inequality. More generally, we call (1.1) the *τ-relaxed triangle inequality*, or *τ-inequality* for short. The smallest possible parameter τ for a given problem instance (with $n = kp$ vertices) can be computed in $O(n^3)$ time. When $\tau < 1$ results, the a priori bounds (depending on τ) on the quality of the solutions delivered by the heuristics are usually tighter than the ones employing only the standard (i.e., $\tau = 1$) triangle inequality.

Successive application of the τ-inequality to a sequence of edges $\{u_i, u_{i+1}\} \in E$ $(i = 0, \ldots, m-1)$ leads to the following *iterated τ-inequality* in the case $\tau \geqslant 1$:

$$d(u_0, u_m) \leqslant \tau^{\lceil \log m \rceil} \cdot \sum_{i=0,\ldots,m-1} d(u_i, u_{i+1}), \tag{1.2}$$

where log denotes the logarithm to the base 2, and $\lceil \lambda \rceil$, the "ceiling" of a real number λ, is the smallest integer greater than or equal to λ. In order to verify this inequality by induction, one may assume that $m = 2^s$ (as $\tau \geqslant 1$); then indeed (for $s \geqslant 2$)

$$\begin{aligned} d(u_0, u_{2^s}) &\leqslant \tau(d(u_0, u_{2^{s-1}}) + d(u_{2^{s-1}}, u_{2^s})) \\ &\leqslant \tau \cdot \tau^{s-1} \cdot \sum_{i=0,\ldots,2^s-1} d(u_i, u_{i+1}). \end{aligned}$$

2. Some decomposable cost functions

We now introduce some of the specific decomposable cost functions which will be treated in the remainder of the paper. Our initial motivation for considering these cost functions stems from the application described in Crama et al. [2]. In that application, the cost of a clique should somehow reflect the total distance travelled by the robot-arm of a machine in order to visit all vertices of the clique. The particular robot-arm under study in Crama et al. [2] could visit at most three locations in one so-called placement round. Hence, this industrial setting gives rise to a 3-dimensional assignment problem with decomposable costs. In general however, a robot-arm may have a larger capacity, allowing it to visit k locations in one round. Then, an instance of the k-dimensional assignment problem with decomposable costs arises. Since the order in which the vertices of the clique will eventually be visited is not known in advance, the distance travelled can only be roughly evaluated. This can be done in several ways.

Sum costs. The sum cost function assigns to every clique $X = \{x_0, \ldots, x_{k-1}\}$ a cost equal to the sum of its edge lengths, i.e.:

$$c(X) = \sum_{0 \leqslant i < j \leqslant k-1} d(x_i, x_j). \tag{2.1}$$

Observe that, since every clique contains the same number of vertices, the cost of a clique with respect to the cost function (2.1) is proportional to the average length of the edges induced by the clique.

Sum cost functions are commonly used in the context of graph partitioning problems, such as those mentioned e.g. in Lengauer [10]. Actually, the k-dimensional assignment problem with sum costs can be seen as a special case of the k-dimensional matching problem studied by Feo and Khellaf [4]: given an (arbitrary) graph G with kp vertices, and a nonnegative weight for each edge $\{u, v\}$ of G, find a partition of the vertex set of G into p sets $V_1, \ldots, V_p$ such that the sum of the weights of the edges contained in $\bigcup_i V_i$ is maximized. Feo and Khellaf [4] present heuristics with guaranteed worst-case performance for this k-dimensional problem. But their bounds have little meaning for our problem, due to the fact that we stated it as a minimization rather than a maximization problem.

Star costs. The cost of a clique $X = \{x_0, \ldots, x_{k-1}\}$ with respect to the star cost function is equal to the sum of the edge lengths of a minimum length spanning star of X, i.e.:

$$c(X) = \min\left\{ \sum_{0 \leqslant i \leqslant k-1} d(x_h, x_i) \mid 0 \leqslant h \leqslant k-1 \right\}. \tag{2.2}$$

In the framework of location theory, the vertex x_h realizing the minimum in (2.2) is called the median vertex of the clique.

Tour costs. The tour cost of the clique $X = \{x_0, \ldots, x_{k-1}\}$ is defined as the cost of a traveling salesman tour on the graph induced by X, in other words:

$$c(X) = \min\left\{ \sum_{0 \leqslant i \leqslant k-1} d(x_{\pi(i)}, x_{\pi(i+1)}) \mid \pi \text{ is a permutation of } \{0, 1, \ldots, k-1\} \right\}, \tag{2.3}$$

where integers are read modulo k.

Tree costs. The cost of the clique $X = \{x_0, \ldots, x_{k-1}\}$ with respect to the tree cost function is the cost of a minimum length spanning tree of the complete graph on X, i.e.:

$$c(X) = \min\left\{ \sum_{\{i,j\} \in A} d(x_i, x_j) \mid (X, A) \text{ is a tree} \right\}. \tag{2.4}$$

Observe that, for $k = 3$, the sum and the tour cost functions are identical, as are the star and the tree cost functions. In Crama and Spieksma [3] it is proved that the 3-dimensional assignment problem with sum or star costs is NP-hard, even when d satisfies the triangle inequality.

3. Approximation algorithms and their performance

Suppose from now on that we have to solve an instance of the k-dimensional assignment problem on (V, E) with respect to a decomposable cost function c, depending on a length function d. We denote by M_{opt} an optimal solution of this instance.

We propose in this section various heuristics for this problem, and we state our main results about the quality of the solutions which they produce. Namely, we state theorems of the form: "for all instances of the k-dimensional assignment problem with decomposable costs satisfying such and such assumptions, the heuristic under consideration produces an assignment M such that $c(M) \leqslant \alpha(k,\tau)\cdot c(M_{\text{opt}})$", where $\alpha(k,\tau)$ is an (explicitly given) function of k and τ. We sometime add to this that "the bound is tight", meaning that there exist instances of the problem satisfying the required assumptions and yielding $c(M) = \alpha(k,\tau)\cdot c(M_{\text{opt}})$. Notice that, for the sake of clarity, the theorems are not always stated in the full generality with which they will be proved in subsequent sections.

Our heuristics fall into two classes, namely "hub heuristics", and "recursive heuristics". As we will see, the main ingredient of these procedures consists in solving a sequence of bipartite matching problems on graphs with $2p$ vertices. As an estimate of the complexity of the heuristics, we will therefore use the number of matching problems which they require to solve. Observe that, in particular, we do not explicitly take into account the time required to compute the cost of a clique in some of the heuristics. This is reasonable for all but the tour cost function; and even the cost of a tour can be quickly computed when k is small.

Hub heuristics. We begin with a description of the single-hub heuristic. There is actually one such heuristic for each $h \in \{0,\ldots,k-1\}$; h is then the "hub" of the heuristic. The *single-hub heuristic* with hub h produces an assignment which is composed of minimum length matchings between X_h and all other parts X_i. It proceeds as follows:

Step 1. For each $i \neq h$, let M_{hi} be a minimum length matching between X_h and X_i with respect to the length function d.

Step 2. Return the k-dimensional assignment $M_h := \{\{x_0,\ldots,x_{k-1}\} \mid x_h \in X_h$ and $\{x_h, x_i\} \in M_{hi},\ i = 0,\ldots,k-1,\ i \neq h\}$.

The single-hub heuristic is clearly polynomial: it only requires the solution of $k-1$ bipartite matching problems. Notice also that the heuristic is quite simple minded, as it only depends on d, and not on the specific cost function c built up from d. But in spite of this simplicity, the quality of the solution delivered by the single-hub heuristic cannot be arbitrarily bad, as is attested by the next statement:

Theorem 3.1. *If c is either the sum or the star or the tour or the tree cost function, then $c(M_h) \leqslant (k-1)\cdot c(M_{\text{opt}})$ for every problem instance satisfying the triangle inequality. This bound is tight. (See* (4.2), (5.3), (6.6), *and* (7.2) *below.)*

More precisely, we will establish in Sections 4–7 that, for each of the four cost functions mentioned in Theorem 3.1, there exists a function $\alpha(k,\tau)$ such that $c(M_h) \leqslant \alpha(k,\tau)\cdot c(M_{\text{opt}})$ and $\alpha(k,\tau) \leqslant k-1$ when $\tau \leqslant 1$. The function $\alpha(k,\tau)$ grows (at most) linearly with τ for the sum and the star cost functions.

An easy way of improving the single-hub heuristic is to compute a solution M_h for each possible choice of the hub h, and to retain the best solution thus found. We then obtain the following *multiple-hub heuristic*:

Step 1. For each pair $i, h \in \{0, \ldots, k-1\}$, $i \neq h$, let M_{hi} be a minimum length matching between X_h and X_i with respect to the length function d.

Step 2. For each $h \in \{0, \ldots, k-1)$, let M_h be the assignment delivered by the single-hub heuristic with hub h, that is, $M_h := \{\{x_0, \ldots, x_{k-1}\} | x_h \in X_h$ and $\{x_h, x_i\} \in M_{hi}$, $i = 0, \ldots, k-1$, $i \neq h\}$.

Step 3. Return the assignment $M_H := M_i$, where $c(M_i) = \min\{c(M_h) | 0 \leqslant h \leqslant k-1\}$.

This heuristic requires the solution of $k(k-1)/2$ bipartite matching problems. The upper bound provided by Theorem 3.1 is trivially valid for $c(M_H)$, but even stronger results can be proved. For instance, we will show:

Theorem 3.2. *If* c *is either the sum or the star cost function, then* $c(M_H) \leqslant (2/k)[(k-2)\tau + 1] \cdot c(M_{\text{opt}})$ *for every problem instance. This bound is tight when* $\tau = 1$. (*See* (4.4) *and* (5.6) *below.*)

Observe that, for $\tau \leqslant 1$, the ratio $c(M_H)/c(M_{\text{opt}})$ is bounded by 2. In the terminology of Papadimitriou and Steiglitz [11], this means that the multiple-hub heuristic is a 1-approximation algorithm for the k-dimensional assignment problem with either sum or star costs, when the edge lengths satisfy the triangle inequality (independently of the value of k). On the other hand, if τ is not bounded from above (or $\tau = \infty$), then there exists no polynomial-time ε-approximation algorithm for the k-dimensional assignment problem restricted to sum or star costs, for any fixed $\varepsilon \geqslant 0$ and any $k \geqslant 3$ (unless, of course, P = NP); see Crama and Spieksma [3]. As a matter of fact, our theorem only implies that $c(M_H)/c(M_{\text{opt}}) \leqslant 2\tau$.

Theorem 3.3. *If* c *is either the tour or the tree cost function, then* $c(M_H) \leqslant \frac{1}{2}k \cdot c(M_{\text{opt}})$ *if* k *is even and* $c(M_H) \leqslant \frac{1}{2}(k - 1/k) \cdot c(M_{\text{opt}})$ *if* k *is odd, for every problem instance satisfying the triangle inequality. This bound is tight.* (*See* (6.9) *and* (7.4) *below.*)

A more precise, but much more intricate bound can again be obtained by dropping the assumption that the length function satisfies the triangle inequality, and by explicitly taking into account the parameter τ. What may be more important here is to observe that the ratio $c(M_H)/c(M_{\text{opt}})$ grows linearly with k, just as for the single-hub heuristic. We do not know whether there exists a polynomial-time ε-approximation algorithm, with ε an absolute constant, for the k-dimensional assignment problem restricted to either tour or tree costs and to edge lengths satisfying the triangle inequality.

Recursive heuristics. We propose a single-pass recursive heuristic with respect to a chosen permutation of $(X_0, \ldots, X_{k-1})$, say e.g. $(X_{\sigma(0)}, \ldots, X_{\sigma(k-1)})$. The idea behind this heuristic is again to produce an assignment of (V, E) by solving a sequence of $k-1$ bipartite matching problems. The solution of the $(i-1)$st subproblem produces an i-dimensional assignment between $X_{\sigma(0)}, X_{\sigma(1)}, \ldots,$ and $X_{\sigma(i-1)}$, say N_{i-1}. The ith subproblem consists in extending this partial assignment by computing an optimal bipartite matching between its cliques (regarded as indivisible) and the vertices of $X_{\sigma(i)}$, with respect to suitably defined edge lengths. The length function for the $(i-1)$st subproblem ($i \geq 3$) is defined as follows: for all $X \in N_{i-1}$ and $u \in X_{\sigma(i)}$, the length of $\{X, u\}$ is the cost $c(X \cup \{u\})$ of the "partial" clique $X \cup \{u\}$; notice that this definition only makes sense if the cost function c can be meaningfully extended to subsets of cliques; this is certainly the case for the four cost functions introduced in Section 2 since these are well defined for al $k \geq 3$.

We are now ready for a formal description of the *single-pass recursive heuristic* associated to the permutation $(X_{\sigma(0)}, \ldots, X_{\sigma(k-1)})$:

Step 1. Let $N_0 := X_{\sigma(0)}$ and $i := 1$.

Step 2. For all $X \in N_{i-1}$ and $u \in X_{\sigma(i)}$, let $\delta(X, u) := c(X \cup \{u\})$.

Step 3. Let N be a minimum length matching between N_{i-1} and $X_{\sigma(i)}$ with respect to the length function δ. Let $N_i := \{X \cup \{u\} \mid X \in N_{i-1},\ u \in X_{\sigma(i)}$ with $\{X, u\} \in N\}$.

Step 4. If $i < k-1$, let $i := i+1$ and go to Step 2. Else, continue.

Step 5. Return the assignment $M_\sigma := N_{k-1}$.

The complexity of the single-pass recursive heuristic is comparable to that of the single-hub heuristic since in both cases $k-1$ bipartite matching problems need to be solved. (Notice however that, as mentioned at the beginning of Section 3, the time needed to update the cost of the partial cliques in Step 2 of the single-pass recursive heuristic is not taken into account by this complexity measure.) On the other hand, the recursive heuristic turns out to be much more difficult to analyze than the hub heuristic. In fact, the results to be presented in Section 8 will only have bearing on one of our "special" cost functions, namely the sum cost function. We will establish:

Theorem 3.4. *If c is the sum cost function, then $c(M_\sigma) \leq \frac{1}{2}k \cdot c(M_{\text{opt}})$ for every problem instance satisfying the triangle inequality. (See* (8.5) *and* (8.7) *below.)*

Comparing this result with Theorem 3.1, we see that, in the case of sum costs, the worst-case performance of the single-pass recursive heuristic is better than that of the single-hub heuristic. For $k = 3$, the bound of Theorem 3.4 is known to be valid and tight, both in the case of sum and of star costs [3]; but it is no longer tight for $k \geq 4$, by the next theorem. In particular, we do not even know whether the ratio $c(M_\sigma)/c(M_{\text{opt}})$ really grows linearly in k (although we suspect that it does).

Theorem 3.5. *If c is the sum cost function, then $c(M_\sigma) \leq \frac{13}{7}c(M_{\text{opt}})$ for all instances of the 4-dimensional assignment problem satisfying the triangle inequality. This bound is tight.*

Just as we did for the single-hub heuristic, we can improve the performance of the single-pass recursive heuristic by applying it for each possible permutation of $(X_0, \ldots, X_{k-1})$, and retaining the best solution thus found. This results in the *multiple-pass recursive heuristic*:

Step 1. For each permutation σ of $\{0, \ldots, k-1\}$, let M_σ be the assignment produced by the single-pass recursive heuristic associated to the permutation $(X_{\sigma(0)}, \ldots, X_{\sigma(k-1)})$.

Step 2. Return the assignment $M_R := M_\pi$, where $c(M_\pi) = \min\{c(M_\sigma) | \sigma$ is a permutation of $\{0, \ldots, k-1\}\}$.

The complexity of the multiple-pass recursive heuristic is quite high, since $k!(k-1)/2$ bipartite matching problems have to be solved in Step 1. As for its performance, we will prove:

Theorem 3.6. *If c is the sum cost function, then* $c(M_R) \leqslant 2[(k+1)/(k-1)] \cdot \ln(k+1) \cdot c(M_{\text{opt}})$ *for every problem instance satisfying the triangle inequality.* (*See* (8.14) *and* (8.17) *below.*)

For $k = 3$, it is known that $c(M_R)/c(M_{\text{opt}}) \leqslant 4/3$ and that the result is tight (for both sum and star costs; see Crama and Spieksma [3]). This is the same worst-case performance as for the multiple-hub heuristic (see Theorem 3.2). More generally, in view of the sophistication of the multiple-pass recursive heuristic, one would expect its worst-case ratio to be at least as good as that of the multiple-hub heuristic. Interestingly enough, however, we will show that the worst-case ratio of the multiple-pass recursive heuristic is not better than the worst-case ratio of the multiple-hub heuristic when $\tau = 1$.

Still, one may hope that $c(M_R)/c(M_{\text{opt}})$ be bounded by a constant. The bound of Theorem 3.6 is disappointing in this regard (but maybe is not tight!). We will nevertheless be able to prove that $c(M_R)/c(M_{\text{opt}}) \leqslant \tau/(1-\tau)$ (i.e., a bound independent of k) for all problem instances such that $\tau < 1$ (see (8.15) below).

In the following sections, we proceed with the proofs of our results.

4. Sum costs

We assume here that the cost of a clique $X = \{x_0, \ldots, x_{k-1}\}$ equals the sum of all edge lengths $d(x_i, x_j)$, that is,

$$c(X) = \sum_{0 \leqslant i < j \leqslant k-1} d(x_i, x_j).$$

We first establish an upper bound on the cost of the assignment M_h produced by the single-hub heuristic ($h = 0, \ldots, k-1$), which is composed of minimum length matchings M_{hi} between X_h and all other parts X_i. Applying the τ-inequality

$$d(x_i, x_j) \leqslant \tau(d(x_h, x_i) + d(x_h, x_j)),$$

for all $X \in M_h$ and $i, j \neq h$, we obtain

$$\begin{aligned} c(M_h) &= \sum_{X \in M_h} \sum_{i<j} d(x_i, x_j) \\ &\leqslant \sum_{X \in M_h} \sum_{i=0,\ldots,k-1} [(k-2)\tau + 1] \cdot d(x_h, x_i) \\ &= [(k-2)\tau + 1] \cdot \sum_{i=0,\ldots,k-1} \sum_{X \in M_h} d(x_h, x_i) \\ &\leqslant [(k-2)\tau + 1] \cdot \sum_{i=0,\ldots,k-1} \sum_{Z \in M_{\text{opt}}} d(z_h, z_i) \end{aligned} \tag{4.1}$$

as each M_{hi} is a minimum length matching. The sum of $d(z_h, z_i)$ for $i = 0, \ldots, k-1$ is a trivial lower bound of $c(Z)$, whence

$$c(M_h) \leqslant [(k-2)\tau + 1] \cdot c(M_{\mathrm{opt}}). \tag{4.2}$$

Example 4.1. To show that this bound is tight for all $\tau \geqslant 1$, consider Fig. 1. Let x_{ij}, $j = 0, \ldots, k-1$ denote the jth vertex of X_i, or in other words:

$$X_i = \{x_{ij} \mid j = 0, \ldots, k-1\} \quad \text{for } i = 0, \ldots, k-1.$$

Observe that in this example each X_i has cardinality k, that is $p = k$. Let us now define the length function d. The notation $x_{ij} = x_{rs}$ in Fig. 1 (and in all subsequent figures) indicates that $d(x_{ij}, x_{rs}) = 0$. The length between any two vertices linked by an edge in Fig. 1 is equal to 1. All other lengths are equal to 2τ, for some $\tau \geqslant 1$. Note that the τ-inequality is fulfilled by this distance function. The single-hub heuristic with hub 0 may find the following assignment (where indices are read modulo k):

$$\begin{aligned} M_0 &= \{\{x_{0j}, x_{1,j+1}, x_{2,j+2}, \ldots, x_{k-1,j+k-1}\} \mid j = 0, \ldots, k-1\} \\ &= \{\{x_{00}, x_{11}, \ldots, x_{k-1,k-1}\}, \{x_{01}, x_{12}, \ldots, x_{k-1,0}\}, \\ &\qquad \ldots, \{x_{0,k-1}, x_{10}, \ldots, x_{k-1,k-2}\}\} \end{aligned}$$

resulting in

$$\begin{aligned} c(M_0) &= k[(k-1) + \tau(k-1)(k-2)] \\ &= k(k-1)[(k-2)\tau + 1]. \end{aligned}$$

It is not difficult to see that

$$\begin{aligned} M_{\mathrm{opt}} &= \{\{x_{0j}, x_{1j}, \ldots, x_{k-1,j}\} \mid j = 0, \ldots, k-1\} \\ &= \{\{x_{00}, x_{10}, \ldots, x_{k-1,0}\}, \{x_{01}, x_{11}, \ldots, x_{k-1,1}\}, \\ &\qquad \ldots, \{x_{0,k-1}, x_{1,k-1}, \ldots, x_{k-1,k-1}\}\} \end{aligned}$$

with $c(M_{\mathrm{opt}}) = k(k-1)$ proving that the bound in (4.2) is tight for $\tau \geqslant 1$.

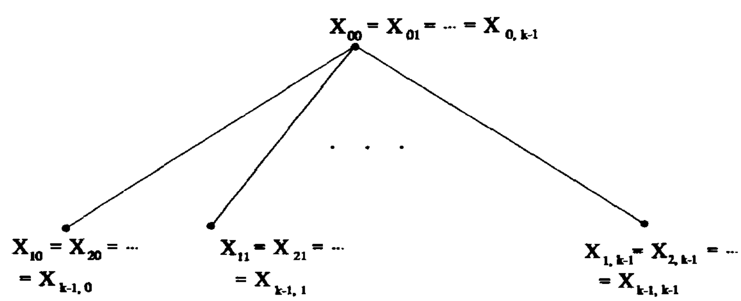

Fig. 1.

The bound (4.2) is certainly not tight when τ is smaller than 1 because $d(z_i, z_j)$ cannot be zero for $i \neq j$. Indeed, using the τ-inequality, we have (given $Z \in M_{\text{opt}}$)

$$d(z_i, z_j) \geq \frac{1}{\tau} \cdot d(z_h, z_i) - d(z_h, z_j)$$

for $i \neq j$ and $i, j \neq h$. Consequently,

$$\sum_{\substack{j=0,\ldots,k-1 \\ j \neq h}} d(z_i, z_j) \geq \frac{k-2}{\tau} \cdot d(z_h, z_i) - \sum_{\substack{j=0,\ldots,k-1 \\ j \neq i}} d(z_h, z_j) \quad \text{for } i \neq h,$$

and further

$$\begin{aligned}\sum_{i,j \neq h} d(z_i, z_j) &\geq \frac{k-2}{\tau} \cdot \sum_{i=0,\ldots,k-1} d(z_h, z_i) - (k-1) \cdot \sum_{j=0,\ldots,k-1} d(z_h, z_j) \\ &\quad + \sum_{i=0,\ldots,k-1} d(z_h, z_i) \\ &= (k-2)\left(\frac{1}{\tau} - 1\right) \cdot \sum_{i=0,\ldots,k-1} d(z_h, z_i).\end{aligned}$$

Hence,

$$\left[(k-2)\left(\frac{1}{\tau} - 1\right) + 2\right] \cdot \sum_{i=0,\ldots,k-1} d(z_h, z_i) \leq 2 \cdot \sum_{i<j} d(z_i, z_j).$$

From this inequality and (4.1) we infer

$$c(M_h) \leq \frac{2((k-2)\tau + 1)}{(k-2)(1/\tau - 1) + 2} \cdot c(M_{\text{opt}}) \quad \text{if } \tau \leq 1. \tag{4.3}$$

Observe that this bound is tight in the cases $\tau = \frac{1}{2}$ (trivially) and $\tau = 1$ (by Example 4.1). It is also tight for all $\tau \leq 1$ when $k = 3$.

Example 4.2. To see this consider Fig. 2. Here, $X_j = \{x_{j0}, x_{j1}, x_{j2}\}$, for $j = 0, 1, 2$. The lengths between pairs of vertices are indicated along the corresponding edges of Fig. 2. Note that the τ-inequality is satisfied if $\tau \leq 1$. Clearly, the optimal solution for this problem equals: $M_{\text{opt}} = \{\{x_{00}, x_{10}, x_{20}\}, \{x_{01}, x_{11}, x_{21}\}, \{x_{02}, x_{12}, x_{22}\}\}$, with cost $c(M_{\text{opt}}) = 3(1 + \tau)$. However, the single-hub heuristic with hub 0 may find the following assignment:

$$M_0 = \{\{x_{00}, x_{11}, x_{22}\}, \{x_{01}, x_{12}, x_{20}\}, \{x_{02}, x_{10}, x_{21}\}\}$$

with

$$c(M_0) = 6(\tau + \tau^2).$$

This results in $c(M_0) = 2\tau \cdot c(M_{\text{opt}})$, proving that (4.3) is tight for all $\tau \leq 1$ when $k = 3$.

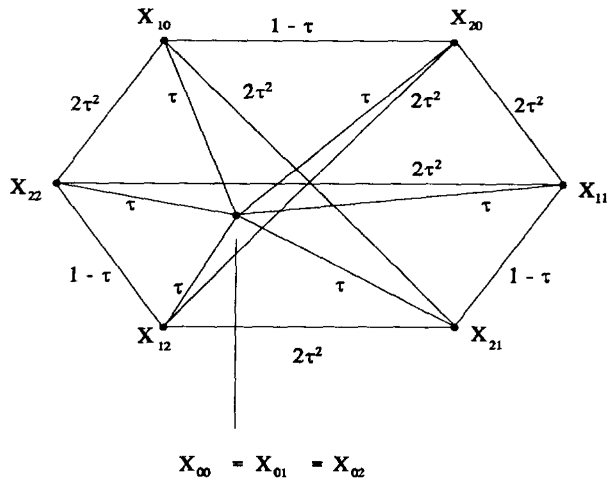

Fig. 2.

Next we derive from (4.1) an upper bound on the cost of the solution returned by the multiple-hub heuristic, i.e., the cost of a best assignment among $M_0, \ldots, M_{k-1}$:

$$\begin{aligned} c(M_H) &= \min_{h=0,\ldots,k-1} c(M_h) \\ &\leqslant \frac{1}{k} \cdot \sum_{h=0,\ldots,k-1} c(M_h) \\ &\leqslant \frac{1}{k} \cdot \sum_{h=0,\ldots,k-1} [(k-2)\tau + 1] \cdot \sum_{i=0,\ldots,k-1} \sum_{Z \in M_{\text{opt}}} d(z_h, z_i) \\ &= \frac{1}{k}[(k-2)\tau + 1] \cdot \sum_{Z \in M_{\text{opt}}} \sum_{i,j} d(z_i, z_j) \\ &= \frac{2}{k}[(k-2)\tau + 1] \cdot c(M_{\text{opt}}). \end{aligned} \tag{4.4}$$

This bound is best possible again when $\tau = \frac{1}{2}$. Observe that

$$\frac{2}{k}[(k-2)\tau + 1] = 2\tau \left[1 - \frac{1}{k}\left(2 - \frac{1}{\tau}\right) \right] < 2\tau \quad \text{if } \tau > \frac{1}{2}.$$

Example 4.3. In order to prove that the bound (4.4) is also tight for $\tau = 1$, consider Fig. 3. The upper vertex in Fig. 3 is labeled by the points $x_{00}, \ldots, x_{k-1,0}$, while the other k vertices are labeled by the points $x_{0j}, x_{1,j+1}, \ldots, x_{k-1,k-1+j}$, expect $x_{k-j,0}$ $(j = 0, \ldots, k-1)$.

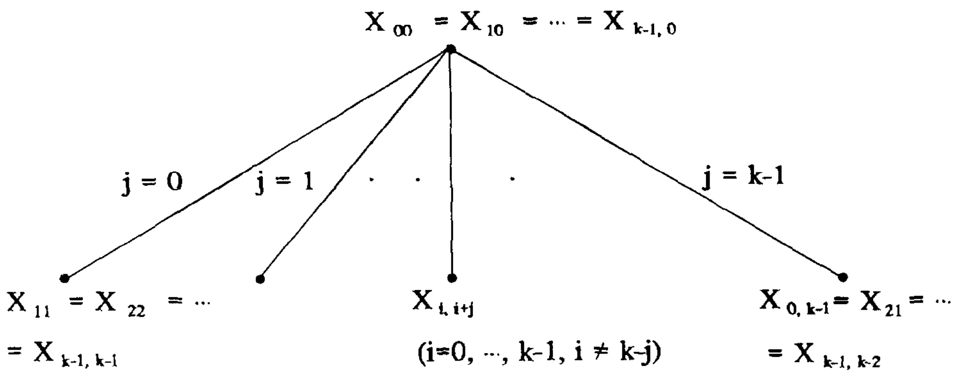

Fig. 3.

The sets $X_0, \ldots, X_{k-1}$ and the length function are defined as for Example 4.1, with $\tau = 1$. Obviously, an optimal assignment is given by

$$M_{\text{opt}} = \{\{x_{i,i+j} | i = 0, \ldots, k-1\} | j = 0, \ldots, k-1\}$$

with

$$c(M_{\text{opt}}) = k(k-1).$$

Since the example is symmetric on $X_0, \ldots, X_{k-1}$, we may assume that the solution returned by the multiple-hub heuristic corresponds to the choice of the hub $h = 0$. Now, for each $i = 1, \ldots, k-1$, an optimal bipartite matching between X_0 and X_i is given by:

$$M_{0i} = \{\{x_{00}, x_{i0}\}, \{x_{0,k-i}, x_{ii}\}\} \cup \{\{x_{0j}, x_{i,i+j}\} | j \neq 0, j \neq k-i\}.$$

The cost of M_{0i} is 2 for each i. The associated k-dimensional assignment $M_H = M_0$ has cost $c(M_H) = 2(k-1)^2$. Thus, $c(M_H) = 2(1 - \frac{1}{k}) \cdot c(M_{\text{opt}})$, proving that the bound (4.4) is tight for $\tau = 1$.

5. Star costs

Now we assume that the cost of a clique $X = \{x_0, \ldots, x_{k-1}\}$ equals the sum of the edge lengths in a minimum length spanning star of X, that is,

$$c(X) = \min\left\{ \sum_{i=0,\ldots,k-1} d(x_h, x_i) | h = 0, \ldots, k-1 \right\}.$$

For every h we obtain

$$\begin{aligned} c(M_h) &\leqslant \sum_{X \in M_h} \sum_{i=0,\ldots,k-1} d(x_h, x_i) \\ &= \sum_{i=0,\ldots,k-1} \sum_{X \in M_h} d(x_h, x_i) \\ &\leqslant \sum_{i=0,\ldots,k-1} \sum_{Z \in M_{\text{opt}}} d(z_h, z_i) \end{aligned} \tag{5.1}$$

because each M_{hi} is a minimum length matching. Assume that for a given $Z \in M_{\text{opt}}$,

$$c(Z) = \sum_{i=0,\dots,k-1} d(z_0, z_i).$$

If $h \neq 0$, then by the τ-inequality we derive

$$\sum_{i=0,\dots,k-1} d(z_h, z_i) \leqslant [(k-2)\tau + 1] \cdot d(z_0, z_h) + \tau \cdot \sum_{\substack{i=0,\dots,k-1 \\ i \neq h}} d(z_0, z_i). \tag{5.2}$$

So, (5.1) and (5.2) yield the estimation

$$c(M_h) \leqslant [(k-2)\tau + 1] \cdot c(M_{\text{opt}}). \tag{5.3}$$

Example 5.1. To prove that this bound is tight for $\tau = 1$, consider the problem instance described in Example 4.1, with $\tau = 1$.

It is easy to see that the optimal solution M_{opt} as well as the solution M_0 found by the single-hub heuristic are identical to the ones described there (remember that the single-hub heuristic does not explicitly take into account the cost function c). With respect to the star costs, we thus have

$$c(M_0) = k(k-1) \quad \text{and} \quad c(M_{\text{opt}}) = k.$$

This proves that the bound derived in (5.3) is tight for $\tau = 1$.

For $\tau < 1$ we can again do better than (5.3). First observe that

$$\frac{1}{\tau} \cdot d(z_0, z_h) \leqslant d(z_h, z_i) + d(z_0, z_i) \quad \text{for } i \neq 0, h,$$

and therefore

$$\left(1 + (k-2)\frac{1}{\tau}\right) \cdot d(z_0, z_h) \leqslant \sum_{i=0,\dots,k-1} d(z_h, z_i) + \sum_{\substack{i=1,\dots,k-1 \\ i \neq h}} d(z_0, z_i).$$

Adding this inequality to (5.2) results in

$$(k-2)(1-\tau) \cdot d(z_0, z_h) \leqslant \tau \cdot \sum_{\substack{i=1,\dots,k-1 \\ i \neq h}} d(z_0, z_i). \tag{5.4}$$

Now, for $\tau \leqslant 1$, multiply the latter inequality by

$$\lambda = \frac{(k-3)\tau + 1}{(k-3)(1-\tau) + 1}$$

and add this to (5.2). Since

$$(k-2)\tau + 1 - \lambda(k-2)(1-\tau) = \tau + \lambda\tau = \frac{(k-1)\tau}{(k-3)(1-\tau) + 1},$$

we conclude that

$$c(M_h) \leqslant \frac{(k-1)\tau}{(k-3)(1-\tau)+1} \cdot c(M_{\text{opt}}) \quad \text{if } \tau \leqslant 1. \tag{5.5}$$

As to an upper bound on the cost of the assignment produced by the multiple-hub heuristic, apply (5.1) and finally the τ-inequality:

$$\begin{aligned} c(M_H) &\leqslant \frac{1}{k} \cdot \sum_{h=0,\dots,k-1} c(M_h) \\ &\leqslant \frac{1}{k} \cdot \sum_{h=0,\dots,k-1} \sum_{i=0,\dots,k-1} \sum_{Z \in M_{\text{opt}}} d(z_h, z_i) \\ &= \frac{2}{k} \cdot \sum_{Z \in M_{\text{opt}}} \sum_{i<j} d(z_i, z_j) \\ &\leqslant \frac{2}{k} \cdot \sum_{Z \in M_{\text{opt}}} [(k-2)\tau + 1] \cdot c(Z) \\ &= \frac{2}{k} [(k-2)\tau + 1] \cdot c(M_{\text{opt}}). \end{aligned} \tag{5.6}$$

Example 5.2. To show that (5.6) is tight for $\tau = 1$, consider again the instance described in Example 4.3. We assume that M_H and M_{opt} remain the same as in Example 4.3 (because $M_H = M_0$ by symmetry of the instance, and M_0 is independent of the cost function c). Here, $c(M_H) = 2(k-1)$ and $c(M_{\text{opt}}) = k$, thus establishing the tightness of (5.6) when $\tau = 1$.

6. Tour costs

The cost of a clique $X = \{x_0, \dots, x_{k-1}\}$ is now defined as the sum of the edge lengths in a minimum length Hamiltonian tour (i.e., spanning cycle) of X:

$$c(X) = \min \left\{ \sum_{i=0}^{k-1} d(x_{\pi(i)}, x_{\pi(i+1)}) \,|\, \pi \text{ is a permutation of } \{0, \dots, k-1\} \right\},$$

where integers i are read modulo k. Assume without loss of generality that for a given X (or later on, given $Z \in M_{\text{opt}}$) the minimum cost is attained for the identity permutation. Then from the $k-2$ inequalities

$$d(x_i, x_{i+1}) \leqslant \tau(d(x_0, x_i) + d(x_0, x_{i+1})) \quad \text{for } i \neq 0,\ k-1$$

we obtain

$$c(X) \leqslant (1+\tau)(d(x_0, x_1) + d(x_0, x_{k-1})) + 2\tau \cdot \sum_{i=2,\dots,k-2} d(x_0, x_i).$$

To simplify our computations, we assume from now on that $\tau \geqslant 1$. Hence

$$c(X) \leqslant 2\tau \cdot \sum_{i=1,\dots,k-1} d(x_0, x_i). \tag{6.1}$$

The analogous inequality holds when 0 is replaced by h. Therefore

$$\begin{aligned} c(M_h) &\leqslant 2\tau \cdot \sum_{i=0,\dots,k-1} \sum_{X \in M_h} d(x_h, x_i) \\ &\leqslant 2\tau \cdot \sum_{i=0,\dots,k-1} \sum_{Z \in M_{\text{opt}}} d(z_h, z_i) \end{aligned} \tag{6.2}$$

since each M_{hi} is a minimum length matching. Applying the iterated τ-inequality (1.2), we get for $2 \leqslant j \leqslant k/2$

$$d(z_0, z_j) \leqslant \tau^{\lceil \log j \rceil} \cdot \sum_{i=0,\dots,j-1} d(z_i, z_{i+1}). \tag{6.3}$$

So, when $1 \leqslant j < k/2$, we have

$$\sum_{i=1,\dots,j} d(z_0, z_i) \leqslant \left(\sum_{i=1,\dots,j} \tau^{\lceil \log i \rceil} \right) \cdot \sum_{i=0,\dots,j-1} d(z_i, z_{i+1})$$

and similarly,

$$\sum_{i=1,\dots,j} d(z_0, z_{k-i}) \leqslant \left(\sum_{i=1,\dots,j} \tau^{\lceil \log i \rceil} \right) \cdot \sum_{i=0,\dots,j-1} d(z_{k-i-1}, z_{k-i}).$$

For k odd, adding up these two inequalities immediately yields

$$\sum_{i=1,\dots,k-1} d(z_0, z_i) \leqslant \left(\sum_{i=1,\dots,(k-1)/2} \tau^{\lceil \log i \rceil} \right) \cdot \sum_{i=0,\dots,k-1} d(z_i, z_{i+1}). \tag{6.4}$$

For k even, one adds one half of the inequality (6.3) with $j = k/2$ to each of the above inequalities and then obtains

$$\begin{aligned} &\sum_{i=1,\dots,k-1} d(z_0, z_i) \\ &\leqslant \left(\frac{1}{2} \tau^{\lceil \log (k/2) \rceil} + \sum_{i=1,\dots,(k-2)/2} \tau^{\lceil \log i \rceil} \right) \cdot \sum_{i=0,\dots,k-1} d(z_i, z_{i+1}). \end{aligned} \tag{6.5}$$

To give a very rough estimate, observe that the sums of powers of τ in (6.4) and (6.5) are bounded above by

$$\frac{k-1}{2} \cdot \frac{1}{\tau} \cdot \tau^{\lceil \log (k-1) \rceil}.$$

Hence from (6.2) we infer

$$c(M_h) \leqslant (k-1) \tau^{\lceil \log (k-1) \rceil} \cdot c(M_{\text{opt}}) \quad \text{for } \tau \geqslant 1. \tag{6.6}$$

This inequality is certainly not tight for $\tau \neq 1$.

Example 6.1. The instance described in Example 4.1 achieves the worst-case bound (6.6) when $\tau = 1$. Indeed, M_0 and M_{opt} are identical to the ones described there, now at cost $c(M_0) = k \cdot 2(k-1)$ and $c(M_{\text{opt}}) = 2 \cdot k$.

From (6.2) one can readily derive a first upper bound of $c(M_H)$, viz.,

$$\begin{aligned} c(M_H) &\leqslant \frac{1}{k} \cdot \sum_{h=0,\dots,k-1} c(M_h) \\ &\leqslant \frac{1}{k} \cdot \sum_{h=0,\dots,k-1} 2\tau \cdot \sum_{i=0,\dots,k-1} \sum_{Z \in M_{\mathrm{opt}}} d(z_h, z_i) \\ &= \frac{4}{k} \tau \cdot \sum_{Z \in M_{\mathrm{opt}}} \sum_{i<j} d(z_i, z_j). \end{aligned} \tag{6.7}$$

In order to estimate the sum of all $d(z_i, z_j)$ in terms of $c(Z)$ we make use of (6.3) (with suitably shifted indices):

$$\sum_{i<j} d(z_i, z_j) \leqslant \begin{cases} \left(\sum_{j=1,\dots,(k-1)/2} j\tau^{\lceil \log j \rceil} \right) \cdot c(Z), & \text{if } k \text{ is odd}, \\ \left(\frac{k}{4} \tau^{\lceil \log (k/2) \rceil} + \sum_{j=1,\dots,(k-2)/2} j\tau^{\lceil \log j \rceil} \right) \cdot c(Z), & \text{if } k \text{ is even} \end{cases}$$

$$\text{for } \tau \geqslant 1. \tag{6.8}$$

The powers of τ occurring in (6.8) are all bounded above by $(1/\tau) \cdot \tau^{\lceil \log (k-1) \rceil}$. So, applying (6.7), we arrive at the following inequality:

$$c(M_H) \leqslant \begin{cases} \frac{1}{2}\left(k - \frac{1}{k}\right) \tau^{\lceil \log (k-1) \rceil} \cdot c(M_{\mathrm{opt}}), & \text{if } k \text{ is odd}, \\ \frac{1}{2} k \tau^{\lceil \log (k-1) \rceil} \cdot c(M_{\mathrm{opt}}), & \text{if } k \text{ is even} \end{cases}$$

$$\text{for } \tau \geqslant 1. \tag{6.9}$$

This bound is certainly not tight when $\tau > 1$; but for $\tau = 1$ it is best possible, as is confirmed by the next example.

Example 6.2. Let us assume that $\tau = 1$ and, for ease of notation, that k is even (the case k odd will be briefly discussed later on).

The instance we consider involves $k\binom{k}{k/2}$ vertices, with $X_j = \{(j, S, \delta) | j \in S \subseteq \{0, \dots, k-1\},\ |S| = k/2,\ \delta \in \{0, 1\}\}$ $(j = 0, \dots, k-1)$. We think of these vertices as being spread over two levels, corresponding respectively to $\delta = 0$ or $\delta = 1$. Within each level, the vertices are clustered into $\binom{k}{k/2}$ groups of cardinality $k/2$; each group contains all vertices (j, S, δ), for a fixed S, and for $j \in S$. The distance between any two vertices in distinct levels is 1, between two vertices in a same group is 0, and between two vertices in a same level, but in different groups, is 2. More formally,

$$\begin{aligned} &d((j, S, 0), (i, T, 1)) = 1 \quad \text{for all } j \in S,\ i \in T, \\ &d((j, S, \delta), (i, S, \delta)) = 0 \quad \text{for all } i, j \in S, \\ &d((j, S, \delta), (i, T, \delta)) = 2 \quad \text{for all } S \neq T,\ j \in S,\ i \in T, \end{aligned}$$

for all $S, T \subseteq \{0, \dots, k-1\}$, $|S| = |T| = k/2$ and for all $\delta \in \{0, 1\}$.

Notice that, for this problem instance, each sum has cost at least 2. The optimal assignment is:

$$M_{\text{opt}} = \left\{\{(j,S,0)\,|\,j\in S\} \cup \{(i,\bar{S},1)\,|\,i\in\bar{S}\}\,\middle|\,|S| = \frac{k}{2}\right\}$$

where $\bar{S} = \{0,\dots,k-1\}\backslash S$. To verify this, observe that each set in M_{opt} is a clique with tour cost equal to 2.

By symmetry, we can assume that $M_H = M_0$. For each $j = 1,\dots,k-1$, an optimal bipartite matching between X_0 and X_j is given by

$$M_{0j} = \left\{\{(0,S,\delta),\,(j,S,\delta)\}\,\middle|\,0,j\in S,\ |S| = \frac{k}{2}\right\}$$

$$\cup \left\{\{(0,S,\delta),\,(j,S\cup\{j\}\backslash\{0\},\,1-\delta)\}\,\middle|\,0\in S,\ j\notin S,\ |S| = \frac{k}{2}\right\}.$$

The total length of this matching is $k/2$; this is certainly optimal, since for each δ such that $j\in S$, $(0,S,\delta)$ must be matched with a vertex at distance at least 1.

Accordingly, the multiple-hub heuristic returns the assignment M_H consisting of the following sum:

$$\{(j,S,\delta)\,|\,j\in S\} \cup \{(j,S\cup\{j\}\backslash\{0\},\,1-\delta)\,|\,j\notin S\} \tag{6.10}$$

for each S such that $0\in S$, $|S| = k/2$, and for $\delta = 0,1$. In words, each clique of M_H contains $k/2$ vertices forming a group in one level, and $k/2$ vertices from $k/2$ distinct groups in the other level. The tour cost of a clique is thus equal to k, i.e., $k/2$ times the cost of a clique in M_{opt}. This implies that $c(M_H) = (k/2)\,c(M_{\text{opt}})$; hence, (6.9) is tight for $\tau = 1$ and k even.

When k is odd, a similar example can be constructed, involving as vertices all triples of the form (j,S,δ) for $j\in S$, $|S|\in\{\lceil k/2\rceil, \lfloor k/2\rfloor\}$ and $\delta\in\{0,1\}$. The definitions of the length function, of M_{opt} and of M_H carry over, if we replace everywhere the condition $|S| = k/2$ by $|S|\in\{\lceil k/2\rceil, \lfloor k/2\rfloor\}$. The tour cost of a clique (6.10) is then

$$2\left\lfloor\frac{k}{2}\right\rfloor \text{ if } |S| = \left\lceil\frac{k}{2}\right\rceil,$$

$$2\left\lceil\frac{k}{2}\right\rceil \text{ if } |S| = \left\lfloor\frac{k}{2}\right\rfloor.$$

This entails, after some computations, that (6.9) is tight when $\tau = 1$ and k is odd. Details are left to the reader.

7. Tree costs

Let now the cost of a clique $X = \{x_0,\dots,x_{k-1}\}$ be the sum of edge lengths in a minimum length spanning tree of X. Then, as in the case of star costs, we obtain

$$c(M_h) \leqslant \sum_{Z\in M_{\text{opt}}}\ \sum_{i=0,\dots,k-1} d(z_h, z_i). \tag{7.1}$$

When $\tau \geqslant 1$, the extended τ-inequality immediately implies:

$$d(z_h, z_i) \leqslant \tau^{\lceil \log(k-1) \rceil} \cdot c(Z),$$

and hence

$$c(M_h) \leqslant (k-1)\tau^{\lceil \log(k-1) \rceil} \cdot c(M_{\text{opt}}) \quad \text{for } \tau \geqslant 1. \tag{7.2}$$

Also

$$c(M_H) \leqslant \frac{2}{k} \cdot \sum_{Z \in M_{\text{opt}}} \sum_{i<j} d(z_i, z_j) \tag{7.3}$$

follows from (7.1). The maximum number of pairs $i < j$ for which an edge of the minimum length spanning tree T of a clique Z lies on a path from z_i to z_j in T equals $(k/2)^2$ if k is even and else equals $[(k+1)/2] \cdot [(k-1)/2]$. So, by (1.2) and (7.3), the following rough estimate can be established:

$$c(M_H) \leqslant \begin{cases} \dfrac{1}{2}\left(k - \dfrac{1}{k}\right)\tau^{\lceil \log(k-1) \rceil} \cdot c(M_{\text{opt}}), & \text{if } k \text{ is odd,} \\ \dfrac{1}{2} k \tau^{\lceil \log(k-1) \rceil} \cdot c(M_{\text{opt}}), & \text{if } k \text{ is even.} \end{cases} \tag{7.4}$$

As we show now, the bounds provided by (7.2) and (7.4) are actually tight for $\tau = 1$:

Example 7.1. For (7.2) consider again the instance described in Example 4.1. The tree costs of the optimal and of the heuristic solutions are the same as their star costs, i.e., $c(M_{\text{opt}}) = k$, $c(M_0) = k(k-1)$, thus establishing the tightness of (7.2) for $\tau = 1$.

As for (7.4), we refer to the instance given in Example 5.2 and we assume that k is even. The optimal and heuristic solutions remain unchanged. The tree cost of a clique in M_{opt} is 1, while the cost of a clique in M_H is $k/2$. This shows that (7.4) is tight for $\tau = 1$ and k even. The analysis is similar for k odd.

8. Recursive heuristics

In this section we establish the worst-case results for the recursive heuristics described in Section 3. As announced there, we only handle here the case of sum costs, i.e., the cost of a clique $X = \{x_0, x_1, \ldots, x_{k-1}\}$ equals the sum of all edge lengths $d(x_i, x_j)$.

First, we verify the upper bound stated in Theorem 3.4 for the worst-case performance of M_σ. Then, we further refine our analysis for different values of τ. Finally, we prove similar results for M_R. Note that these results are in fact generalizations of those results given in Crama and Spieksma [3], where the case $k = 3$, $\tau = 1$ is dealt with.

Assume without loss of generality that σ is the identity permutation. We will also make the convention that, whenever we write a clique X as $\{x_0, \ldots, x_{k-1}\}$, then $x_j \in X_j$

$(j = 0, \ldots, k-1)$. Let us recursively define a function $\alpha(k, \tau)$ as follows:

$$\alpha(2,\tau) = 1,$$

$$\alpha(k,\tau) = \frac{(k-1+2\tau)(k-2)\tau}{((k-2)\tau+1)(k-1)} \cdot \alpha(k-1,\tau) + \frac{1}{k-1} \quad \text{for } k \geqslant 3. \tag{8.1}$$

We will prove by induction on k that $\alpha(k,\tau)$ is an upper bound for the ratio $c(M_\sigma)/c(M_{\text{opt}})$. For $h = 0, \ldots, k-2$ let

$$M_\sigma^h = \{\{x_0, \ldots, x_{k-2}, x_{k-1}\} \mid \{x_0, \ldots, x_{k-2}\} \subseteq Y \text{ for some } Y \in M_\sigma \text{ and } \{x_h, x_{k-1}\} \subseteq Z \text{ for some } Z \in M_{\text{opt}}\}.$$

Thus the cliques in M_σ^h coincide on $X_0 \cup \cdots \cup X_{k-2}$ with the cliques of M_σ, and on $X_h \cup X_{k-1}$ with the cliques of M_{opt}. Then an upper bound for the cost of M_σ is given by the cost of each M_σ^h, viz.

$$c(M_\sigma) \leqslant \sum_{X \in M_\sigma} \sum_{i<j\leqslant k-2} d(x_i,x_j) + \sum_{X \in M_\sigma^h} \sum_{i=0,\ldots,k-2} d(x_i,x_{k-1}). \tag{8.2}$$

Using the τ-inequality $d(x_i,x_{k-1}) \leqslant \tau(d(x_i,x_h) + d(x_h,x_{k-1}))$ for each $i \neq h$, we obtain

$$\begin{aligned} &\sum_{X \in M_\sigma^h} \sum_{i=0,\ldots,k-2} d(x_i,x_{k-1}) \\ &\leqslant \sum_{X \in M_\sigma^h} \left[d(x_h,x_{k-1}) + \tau(k-2) \cdot d(x_h,x_{k-1}) + \tau \cdot \sum_{i=0,\ldots,k-2} d(x_i,x_h) \right] \\ &= [(k-2)\tau+1] \cdot \sum_{Z \in M_{\text{opt}}} d(z_h,z_{k-1}) + \tau \cdot \sum_{X \in M_\sigma} \sum_{i=0,\ldots,k-2} d(x_i,x_h). \end{aligned} \tag{8.3}$$

Combining this inequality with (8.2) and summing over $h = 0, \ldots, k-2$ yields

$$\begin{aligned} (k-1) \cdot c(M_\sigma) &\leqslant (k-1) \cdot \sum_{X \in M_\sigma} \sum_{i<j\leqslant k-2} d(x_i,x_j) \\ &\quad + 2\tau \cdot \sum_{X \in M_\sigma} \sum_{i<h\leqslant k-2} d(x_i,x_h) \\ &\quad + [(k-2)\tau+1] \cdot \sum_{Z \in M_{\text{opt}}} \sum_h d(z_h,z_{k-1}) \\ &= (k-1+2\tau) \cdot \sum_{X \in M_\sigma} \sum_{i<j\leqslant k-2} d(x_i,x_j) \\ &\quad + [(k-2)\tau+1] \cdot \sum_{Z \in M_{\text{opt}}} \sum_j d(z_j,z_{k-1}). \end{aligned}$$

By induction, we derive

$$\begin{aligned} &(k-1) \cdot c(M_\sigma) \\ &\leqslant (k-1+2\tau) \cdot \alpha(k-1,\tau) \cdot \sum_{Z \in M_{\text{opt}}} \sum_{i<j\leqslant k-2} d(z_i,z_j) \\ &\quad + [(k-2)\tau+1] \cdot \sum_{Z \in M_{\text{opt}}} \sum_{j=0,\ldots,k-2} d(z_j,z_{k-1}). \end{aligned} \tag{8.4}$$

In view of this, and using the τ-inequality

$$d(z_i, z_j) \leqslant \tau(d(z_i, z_{k-1}) + d(z_j, z_{k-1})) \quad \text{for } i < j \leqslant k-2,$$

we see that (8.4) entails

$$\begin{aligned}(k-1)\cdot c(M_\sigma) &\leqslant [(k-1+2\tau)\cdot\alpha(k-1,\tau) - \lambda]\cdot \sum_{Z\in M_{\text{opt}}} \sum_{i<j\leqslant k-2} d(z_i, z_j) \\ &\quad + [(k-2)\tau + 1 + (k-2)\tau\lambda]\cdot \sum_{Z\in M_{\text{opt}}} \sum_j d(z_j, z_{k-1})\end{aligned}$$

for all $\lambda \geqslant 0$. Choose $\lambda = (k-1+2\tau)\cdot\alpha(k-1,\tau)/((k-2)\tau+1) - 1$. From (8.1), one can prove by induction on k that the following inequality holds for all $k \geqslant 3$ and $\tau \geqslant \frac{1}{2}$:

$$(k-1+2\tau)\cdot\alpha(k-1,\tau) \geqslant (k-2)\tau + 1.$$

Hence $\lambda \geqslant 0$, and we arrive at:

$$(k-1)\cdot c(M_\sigma) \leqslant \left[\frac{(k-1+2\tau)(k-2)\tau}{(k-2)\tau+1}\cdot\alpha(k-1,\tau) + 1\right]\cdot c(M_{\text{opt}}),$$

or equivalently

$$c(M_\sigma) \leqslant \alpha(k,\tau)\cdot c(M_{\text{opt}}). \tag{8.5}$$

In the sequel, we discuss the behavior of the function $\alpha(k,\tau)$. First observe that $\alpha(k,\frac{1}{2}) = 1$, as one would expect. More generally, if $\frac{1}{2} \leqslant \tau < \frac{1}{2}\sqrt{2}$, we can prove that $\alpha(k,\tau)$ is bounded above by a constant only depending on τ. To see this, define:

$$\gamma(k,\tau) = \frac{(k-2)\tau + 1}{k(1-2\tau^2) + 4\tau^2 - 1}.$$

It is not hard to prove that, for $\frac{1}{2} \leqslant \tau < \frac{1}{2}\sqrt{2}$ and $k \geqslant 3$,

$$\gamma(k-1,\tau) \leqslant \gamma(k,\tau) \leqslant \frac{\tau}{1-2\tau^2},$$

and

$$\gamma(k,\tau) = \frac{(k-1+2\tau)(k-2)\tau}{[(k-2)\tau+1](k-1)}\cdot\gamma(k,\tau) + \frac{1}{k-1}.$$

From these observations and from (8.1), it then follows by induction that

$$\alpha(k,\tau) \leqslant \gamma(k,\tau) \leqslant \frac{\tau}{1-2\tau^2} \tag{8.6}$$

for all $k \geqslant 3$ and for $\frac{1}{2} \leqslant \tau < \frac{1}{2}\sqrt{2}$. For $\tau = \frac{1}{2}\sqrt{2}$, we can show that $\alpha(k, \frac{1}{2}\sqrt{2})$ grows logarithmically with k. Indeed, define $\xi(k) = [(k-1)/(k-1+\sqrt{2})] \cdot \alpha(k, \frac{1}{2}\sqrt{2})$ and rewrite (8.1), as

$$\xi(2) = \frac{1}{1+\sqrt{2}},$$

$$\xi(k) = \xi(k-1) + \frac{1}{k-1+\sqrt{2}} \quad \text{for all } k \geqslant 3.$$

Solving this difference equation yields the following expression for $\alpha(k, \frac{1}{2}\sqrt{2})$:

$$\alpha(k, \tfrac{1}{2}\sqrt{2}) = \left(1 + \frac{\sqrt{2}}{k-1}\right) \cdot \sum_{j=2}^{k} \left(\frac{1}{j+\sqrt{2}-1}\right).$$

The inequalities

$$\sum_{j=2,\ldots,k} \frac{1}{j} < \ln k < \sum_{j=1,\ldots,k-1} \frac{1}{j} \quad \text{for all } k > 1$$

immediately imply the logarithmic behavior of $\alpha(k, \frac{1}{2}\sqrt{2})$. Hence, also $\alpha(k, \tau)$ grows at least logarithmically for $\tau \geqslant \frac{1}{2}\sqrt{2}$.

For $\tau = 1$, the reader will easily verify by substitution into (8.1) that

$$\alpha(k, 1) = \frac{1}{2}k. \tag{8.7}$$

Hence, when the "usual" triangle inequality holds, the upper bound for the worst-case ratio of M_σ is linear in k.

Let us now show that $\alpha(k, \tau)$ is a tight upper bound on $c(M_\sigma)/c(M_{\text{opt}})$ when $k = 3$, for all $\tau \geqslant \frac{1}{2}$.

Example 8.1. Let $X_j = \{x_{j0}, x_{j1}\}$ for $j = 0, 1, 2$, and let the edge lengths be defined as indicated on Fig. 4. Note that the τ-inequality is satisfied. With $\sigma = (0, 1, 2)$, we may

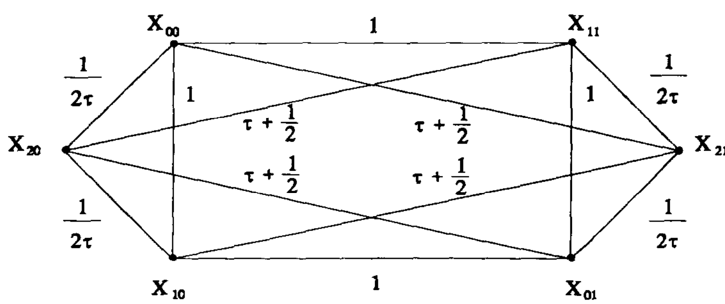

Fig. 4.

obtain: $M_\sigma = \{\{x_{00}, x_{11}, x_{20}\}, \{x_{01}, x_{10}, x_{21}\}\}$ with $c(M_\sigma) = 2(1\frac{1}{2} + \tau + 1/(2\tau))$. Here $M_{\mathrm{opt}} = \{\{x_{00}, x_{10}, x_{20}\}, \{x_{01}, x_{11}, x_{21}\}\}$ with $c(M_{\mathrm{opt}}) = 2(1 + 1/\tau)$. This results in

$$c(M_\sigma)/c(M_{\mathrm{opt}}) = \tau + \frac{1}{2} = \alpha(3, \tau).$$

Unfortunately, the upper bound $\alpha(k, \tau)$ is not tight when $k > 3$, as is asserted by Theorem 3.5 for $k = 4$. To prove Theorem 3.5, assume w.l.o.g. $\sigma = (0, 1, 2, 3)$. Combining (8.2) and (8.3) for $k = 4$ and $\tau = 1$, we obtain for $h = 0, 1, 2$ respectively:

$$c(M_\sigma) \leqslant \sum_{X \in M_\sigma} (d(x_1, x_2) + 2d(x_0, x_1) + 2d(x_0, x_2)) + 3 \sum_{Z \in M_{\mathrm{opt}}} d(z_0, z_3), \tag{8.8}$$

$$c(M_\sigma) \leqslant \sum_{X \in M_\sigma} (d(x_0, x_2) + 2d(x_0, x_1) + 2d(x_1, x_2)) + 3 \sum_{Z \in M_{\mathrm{opt}}} d(z_1, z_3), \tag{8.9}$$

$$c(M_\sigma) \leqslant \sum_{X \in M_\sigma} (d(x_0, x_1) + 2d(x_0, x_2) + 2d(x_1, x_2)) + 3 \sum_{Z \in M_{\mathrm{opt}}} d(z_2, z_3). \tag{8.10}$$

Multiplying (8.8) and (8.9) by 4 and (8.10) by 13 and adding up results in

$$\begin{aligned} 21c(M_\sigma) \leqslant{} & \sum_{X \in M_\sigma} (29d(x_0, x_1) + 38d(x_0, x_2) + 38d(x_1, x_2)) \\ & + \sum_{Z \in M_{\mathrm{opt}}} (12d(z_0, z_3) + 12d(z_1, z_3) + 39d(z_2, z_3)). \end{aligned} \tag{8.11}$$

On the other hand, we can also use (8.2) and (8.3) with $k = 3$ and $h = 0, 1$ in order to derive

$$\sum_{X \in M_\sigma} (d(x_0, x_2) + d(x_1, x_2)) \leqslant 2 \cdot \sum_{Z \in M_{\mathrm{opt}}} d(z_0, z_2) + \sum_{X \in M_\sigma} d(x_0, x_1),$$

$$\sum_{X \in M_\sigma} (d(x_0, x_2) + d(x_1, x_2)) \leqslant 2 \cdot \sum_{Z \in M_{\mathrm{opt}}} d(z_1, z_2) + \sum_{X \in M_\sigma} d(x_0, x_1).$$

Multiplying both inequalities by 19, further the inequality

$$\sum_{X \in M_\sigma} d(x_0, x_1) \leqslant \sum_{Z \in M_{\mathrm{opt}}} d(z_0, z_1)$$

by 67, and then adding all these to (8.11), we obtain

$$\begin{aligned} 21c(M_\sigma) \leqslant \sum_{Z \in M_{\mathrm{opt}}} [&67d(z_0, z_1) + 38d(z_0, z_2) + 12d(z_0, z_3) \\ &+ 38d(z_1, z_2) + 12d(z_1, z_3) + 39d(z_2, z_3)]. \end{aligned}$$

Combining this with the triangle inequalities

$$d(z_0, z_1) \leqslant d(z_0, z_2) + d(z_1, z_2),$$

$$27d(z_0, z_1) \leqslant 27d(z_0, z_3) + 27d(z_1, z_3)$$

leads to $c(M_\sigma) \leqslant \frac{13}{7}c(M_{\mathrm{opt}})$, as required for Theorem 3.5.

The following example shows that this bound is best possible.

Example 8.2. Let $X_j = \{x_{j0}, x_{j1}, x_{j2}\}$ for $j = 0, 1, 2, 3$. In order to define the length function, consider the graph G of Fig. 5. The numbers along the edges of G indicate the length of these edges. For those pairs of vertices u, v for which no edge is drawn, $d(u, v)$ is equal to the length of the shortest path between u and v in G. It is not difficult to see that the optimal solution is $M_{\mathrm{opt}} = \{\{x_{0j}, x_{1j}, x_{2j}, x_{3j}\} \mid j = 0, 1, 2\}$, with $c(M_{\mathrm{opt}}) = 21$. For $\sigma = (0, 1, 2, 3)$, we get $M_{01} = \{\{x_{00}, x_{12}\}, \{x_{01}, x_{10}\}, \{x_{02}, x_{11}\}\}$. Next, we obtain the 3-dimensional assignment $M = \{\{x_{00}, x_{12}, x_{21}\}, \{x_{01}, x_{10}, x_{22}\}, \{x_{02}, x_{11}, x_{20}\}\}$. All matchings between the cliques of M and X_3 have the same cost, namely 39. Hence, $c(M_\sigma)/c(M_{\mathrm{opt}}) = \frac{39}{21} = \frac{13}{7}$.

Let us finally handle the case of the multiple-pass recursive heuristic and the proof of Theorem 3.6. Define

$$\beta(2, \tau) = 1,$$

$$\beta(k, \tau) = \frac{(k - 1 + 2\tau)(k - 2)}{k(k - 1)} \cdot \beta(k - 1, \tau) + \frac{2[(k - 2)\tau + 1]}{k(k - 1)}. \tag{8.12}$$

We want to prove by induction on k that $\beta(k, \tau)$ is an upper bound on $c(M_R)/c(M_{\mathrm{opt}})$.

Consider an instance of the k-dimensional assignment problem, and let M_{opt} be its optimal solution. Assume without loss of generality that, when applying the multiple-pass recursive heuristic to the $(k - 1)$-dimensional subproblem obtained by

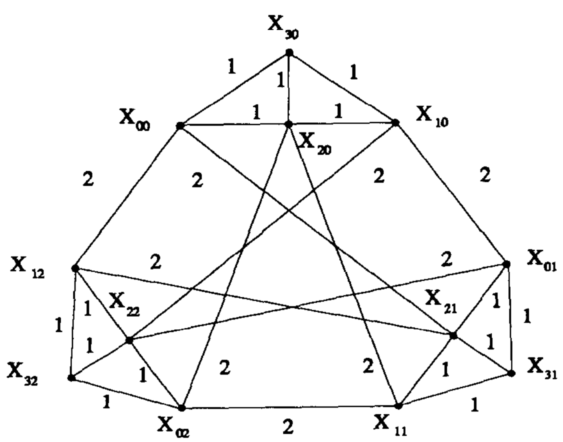

Fig. 5.

disregarding X_{k-1}, the best solution is produced for the permutation $(0, 1, \ldots, k-2)$ of the indices. Let now $\sigma = (0, 1, \ldots, k-1)$, and observe that, by the induction hypothesis,

$$\sum_{X \in M_\sigma} \sum_{i<j \leqslant k-2} d(x_i, x_j) \leqslant \beta(k-1, \tau) \cdot \sum_{Z \in M_{\mathrm{opt}}} \sum_{i<j \leqslant k-2} d(z_i, z_j).$$

So, in the same way as we earlier derived (8.4), we now obtain:

$$\begin{aligned}(k-1) \cdot c(M_R) \leqslant (k-1+2\tau) \cdot \beta(k-1, \tau) \cdot \sum_{Z \in M_{\mathrm{opt}}} \sum_{i<j \leqslant k-2} d(z_i, z_j) \\ + [(k-2)\tau + 1] \cdot \sum_{Z \in M_{\mathrm{opt}}} \sum_j d(z_j, z_{k-1}).\end{aligned} \tag{8.13}$$

Moreover, the reasoning leading to (8.13) can be repeated $k-1$ times, by disregarding $X_0, \ldots, X_{k-2}$ instead of X_{k-1}. This yields the following inequality, for $h = 0, 1, \ldots, k-1$:

$$\begin{aligned}&(k-1) \cdot c(M_R) \\ &\quad \leqslant (k-1+2\tau) \cdot \beta(k-1, \tau) \cdot \sum_{Z \in M_{\mathrm{opt}}} \sum_{\substack{0 \leqslant i<j \leqslant k-1 \\ i,j \neq h}} d(z_i, z_j) \\ &\quad\quad + [(k-2)\tau + 1] \cdot \sum_{Z \in M_{\mathrm{opt}}} \sum_j d(z_j, z_h).\end{aligned}$$

Adding these up, we get:

$$\begin{aligned}&k(k-1) \cdot c(M_R) \\ &\quad \leqslant ((k-1+2\tau)(k-2) \cdot \beta(k-1, \tau) + 2[(k-2)\tau + 1]) \cdot c(M_{\mathrm{opt}}),\end{aligned}$$

or, in view of (8.12)

$$c(M_R) \leqslant \beta(k, \tau) \cdot c(M_{\mathrm{opt}}). \tag{8.14}$$

Let us now investigate the behavior of $\beta(k, \tau)$ for various values of τ. First, when $\frac{1}{2} \leqslant \tau < 1$, one has

$$\beta(k, \tau) \leqslant \eta(k, \tau) \stackrel{\text{def}}{=} \frac{(k-2)\tau + 1}{k - (k-2)\tau - 1} \leqslant \frac{\tau}{1-\tau}. \tag{8.15}$$

To see this, notice that $\eta(k, \tau) \geqslant \eta(k-1, \tau)$ and

$$\eta(k, \tau) = \frac{(k-1+2\tau)(k-2)}{k(k-1)} \cdot \eta(k, \tau) + \frac{2[(k-2)\tau + 1]}{k(k-1)}$$

when $\frac{1}{2} \leqslant \tau < 1$, and use the definition (8.12).

When $\tau = 1$, $\beta(k, 1)$ grows logarithmically with k. Indeed, substituting $\tau = 1$ in (8.12) and defining

$$\zeta(k) = \frac{k-1}{k+1} \cdot \beta(k, 1),$$

we arrive at

$$\zeta(k) = \zeta(k-1) + \frac{2(k-1)}{k(k+1)}.$$

This equation is easily solved, and leads to

$$\beta(k, 1) = \frac{2(k+1)}{k-1} \cdot \sum_{j=1}^{k} \frac{1}{j+1} - \frac{2k}{k-1}. \tag{8.16}$$

For $k \geqslant 2$, define

$$C_k = \sum_{j=1}^{k} \frac{1}{j} - \ln k.$$

It is well known that $1 > C_2 > C_3 > \cdots > C > \frac{1}{2}$ and $\lim_{k \to \infty} C_k = C$, where $C = 0.57721\ldots$ is the Euler–Mascheroni constant. Substituting in (8.16), we thus conclude that

$$\frac{2(k+1)}{k-1}\left[\ln(k+1) - \frac{1}{2}\right] - \frac{2k}{k-1} < \beta(k, 1) < \frac{2(k+1)}{k-1}\ln(k+1) - \frac{2k}{k-1}. \tag{8.17}$$

As mentioned in Section 3, we do not know whether $\beta(k, \tau)$ is a tight upper bound on $c(M_R)/c(M_{\text{opt}})$, except when $\tau = \frac{1}{2}$, or $k = 3$ and $\tau = 1$ (see Crama and Spieksma [3]). In fact, all we can show is that, in the worst case, $c(M_R)/c(M_{\text{opt}})$ grows at least like $2 - 2/k$ when $\tau = 1$ (notice that this is exactly the worst-case ratio of the multiple-hub heuristic; see Section 4). We demonstrate this claim with the following example.

Example 8.3. Consider the instance described in Example 4.3. As this example is symmetric, it suffices to apply the single-pass recursive heuristic to it. The reader may check by induction that it is possible for this heuristic to find the same solution found by the multiple-hub heuristic.

Acknowledgement

We thank Ir. Hans van der Stel for his assistance in solving some of the difference equations arising in this work. The second author was partially supported in the course of this research by AFOSR grants 89-0512 and 90-0008 and an NSF grant STC 88-09648 to Rutgers University.

References

[1] E. Balas and M.J. Saltzman, Facets of the three-index assignment polytope, Discrete Appl. Math. 23 (1989) 201–229.

[2] Y. Crama, A.W.J. Kolen, A.G. Oerlemans and F.C.R. Spieksma, Throughput rate optimization in the automated assembly of printed circuits boards, Ann. Oper. Res. 26 (1990) 455–480.

[3] Y. Crama and F.C.R. Spieksma, Approximation algorithms for three-dimensional assignment problems with triangle inequalities, European J. Oper. Res. 60 (1992) 273–279.

[4] T.A. Feo and M. Khellaf, A class of bounded approximation algorithms for graph partitioning, Networks 20 (1990) 181–195.

[5] A.M. Frieze, A bilinear programming formulation of the 3-dimensional assignment problem, Math. Programming 7 (1974) 376–379.

[6] A.M. Frieze and J. Yadegar, An algorithm for solving 3-dimensional assignment problems with application to scheduling a teaching practice, J. Oper. Res. Soc. 32 (1981) 989–995.

[7] K.B. Haley, The multi-index problem, Oper. Res. 11 (1963) 368–379.

[8] P. Hansen and L. Kaufman, A primal-dual algorithm for the three-dimensional assignment problem, Cahiers Centre Études Rech. Opér. 15 (1973) 327–336.

[9] R.M. Karp, Reducibility among combinatorial problems, in: R.E. Miller and J.W. Thatcher, eds., Complexity of Computer Computations (Plenum Press, New York, 1972) 85–103.

[10] T. Lengauer, Combinatorial Algorithms for Integrated Circuit Layout (Wiley, New York; Teubner, Stuttgart, 1990).

[11] C.H. Papadimitriou and K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity (Prentice-Hall, Englewood Cliffs, NJ, 1982).

[12] W.P. Pierskalla, The multidimensional assignment problem, Oper. Res. 16 (1968) 422–431.