Available online at www.sciencedirect.com
SCIENCE DIRECT®
Information Processing Letters 97 (2006) 203–207
Information Processing Letters
www.elsevier.com/locate/ipl

A polynomial time approximation scheme for embedding a directed hypergraph on a ring

Kang Li [a], Lusheng Wang [b,*]

[a] *School of Information Science and Engineering, Shandong University, Jinan, Shandong Province, P.R. China*
[b] *Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong*

Received 17 March 2005; received in revised form 29 September 2005; accepted 27 October 2005

Available online 17 November 2005

Communicated by F.Y.L. Chin

Abstract

We study the problem of embedding a directed hypergraph on a ring that has applications in optical network communications. The undirected version (MCHEC) has been extensively studied. It was shown that the undirect version was NP-complete. A polynomial time approximation scheme (PTAS) for the undirected version has been developed. In this paper, we design a polynomial time approximation scheme for the directed version.

Keywords: Approximation algorithms; Directed hyperedges embedding; Rings

1. Introduction

The problem of embedding hyperedges on a ring was originally proposed for electronic design automation, where the objective is to route within a minimum-area rectangle [9,1]. The problem of embedding undirected hyperedges on a ring with minimum congestion (MCHEC) has applications in parallel computing as well as multicast routing. Here we study the problem of embedding directed hyperedges on a ring. It models the case where the links in the network is directed.

For the undirected case (MCHEC), Ganley and Cohoon in [2] proved that the problem was NP-hard and gave a ratio-3 approximation algorithm. They also gave an algorithm that can solve the case where the congestion is a constant. Several ratio-2 approximation algorithms were given in [3,5]. Gu and Wang presented a ratio-1.8 approximation algorithm [4]. Recently, Deng and Li proposed a polynomial time approximation scheme (PTAS) for the problem. Their approach comes from the techniques for string problems [6].

In this paper, we study the problem of embedding directed hyperedges on a ring. We extend the method in [8] to get a polynomial time approximation scheme for the directed version. We have developed a new technique for the proofs of some key lemmas. This technique can also be applied to the undirected case. The new proofs allow us to reduce the time complexity of the algorithms in [8] by a factor of $O(m)$, where m is the total number of hyperedges.

* Corresponding author.
E-mail addresses: kangli@sdu.edu.cn (K. Li), lwang@cs.cityu.edu.hk (L. Wang).

0020-0190/$ – see front matter
doi:10.1016/j.ipl.2005.10.008

The paper is organized as follows: We first give some definitions in Section 2. Section 3 deals with a special case, where the number of hyperedges m is $O(\log n)$. Section 4 gives an algorithm that solves the case where the number of hyperedges m is of $O(C_{\text{opt}})$, and c_{opt} is the minimum congestion cost of an optimal embedding. The general case is solved in Section 5.

2. Preliminaries

A *ring* of n nodes is a directed graph $R = (V, E_R)$, where $V = \{1, 2, \ldots, n\}$ is the set of n vertices on the ring and $E_R = \{e_i^+ = (i, i+1), e_i^- = (i+1, i) \mid i = 1, 2, \ldots, n\}$ is the set of $2n$ directed edges on the ring, where $n+1$ is treated as 1. Consider the same set of vertices $V = \{1, 2, \ldots, n\}$. A directed hyperedge $h = (u, S)$ is a pair, where $u \in V$ is the source of the hyperedge and $S \subseteq V - \{u\}$ is the set of sinks. In communication applications, each hyperedge h represents a request that asks to send a message from u to every vertex in S. Let $H = (V, E_H)$ be a directed hypergraph with the same set of vertices V and a set of m directed hyperedges $E_H = \{h_1, h_2, \ldots, h_m\}$.

Let $h_j = (u_j, S_j)$ be a hyperedge in E_H, where $S_j = \{i_1^j, i_2^j, \ldots, i_{k_j}^j\}$. $k_j = |S_j|$ denotes the total number of sink vertices in the hyperedge. For convenience, we use i_0^j to denote u_j. Assume that the $k_j + 1$ vertices $i_0^j, i_1^j, \ldots, i_{k_j}^j$ follow the clockwise order on the ring. P_k^j denotes the segment of vertices on the ring from vertex i_k^j to vertex i_{k+1}^j for $k = 0, 1, \ldots, k_j - 1$ and $P_{k_j}^j$ denotes the segment of vertices on the ring from vertex $i_{k_j}^j$ to vertex i_0^j. In order to realize the request h_j on the ring, one can cut one of the paths P_k for $k = 0, 1, \ldots, k_j$ and obtain two directed paths on the ring both starting from i_0^j. This forms an embedding of h_j on the ring. For a hyperedge h_j, there are $k_j + 1$ different embeddings, one for each P_k^j (by cutting P_k^j). An *embedding* of h_j is a P_k^j *embedding* if P_k^j is cut.

Given an embedding x of all the hyperedges E_H, the *congestion* $e_i^+(x)$ or $e_i^-(x)$ of a directed edge e_i^+ or e_i^- is the number of times that the edge e_i^+ or e_i^- is used in the embedding. When x is clear, we also use $c(e_i^+)$ and $c(e_i^-)$ to denote the congestion. The problem here is to find an embedding for each $h_j \in E_H$ such that every edge e_i^+ and e_i^- on the ring is used at most c times and c is minimized. We refer the problem as the *embedding directed hyperedges on the ring* problem (EDHR, for short).

3. Enumerating method for $m = O(\log n)$

In this section, we give an algorithm with ratio $1 + \frac{1}{r}$ for the case where there are $O(\log n)$ hyperedges in E_H. The basic idea of our algorithm is to choose $2r$ edges on the ring, where r is a constant related to the ratio, and for each hyperedge h_j we only have to cut a P_k^j that contains one of the $2r$ selected edges on the ring. By doing this, for each h_j we only have to consider $2r$ choices (instead of $k_j + 1$ choices for an optimal solution). Since there are at most $O(\log n)$ hyperedges, the time complexity is $O((2r)^{O(\log n)})$, that is polynomial.

Let $i_1, i_2, \ldots, i_{2r}$ be $2r$ indices representing the $2r$ edges $(i_1, i_1+1), (i_2, i_2+1), \ldots, (i_{2r}, i_{2r}+1)$ on the ring R. Let $x = (x_1, x_2, \ldots, x_m)$ be an embedding of H, where x_j indicates the choice P_k^j that is cut for the embedding of h_j. We use $E_j(x)$ to denote the segment P_k^j that is cut for the embedding x of h_j.

Let $Q_{i_1, i_2, \ldots, i_{2r}}(x)$ be a set of indices of hyperedges such that j is in $Q_{i_1, i_2, \ldots, i_{2r}}(x)$ if and only if $E_j(x)$ contains at least one of the $2r$ edges (i_1, i_1+1), (i_2, i_2+1), $\ldots, (i_{2r}, i_{2r}+1)$.

Lemma 1. *Let x be any embedding of H. For any fixed index $1 \leqslant i_1 \leqslant n$, there exist $2r - 1$ indices $i_2, i_3, \ldots, i_{2r}$ such that for any embedding x' satisfying $x'_j = x_j$ for every $j \in Q_{i_1, i_2, \ldots, i_{2r}}(x)$, we have*

$$e_i^+(x') - e_i^+(x) \leqslant \frac{1}{r} e_i^+(x) \quad \textit{and}$$

$$e_i^-(x') - e_i^-(x) \leqslant \frac{1}{r} e_i^-(x)$$

for any directed edge e_i^+ and e_i^- in E_R on the ring.

Proof. We prove the lemma by giving a way to find the $2r - 1$ indices. Let c be the congestion for the embedding x that we want to approximate. First, we select an arbitrary edge, say, $e_1^+ = (1, 2) = (i_1, i_1 + 1)$ on the ring. In the embedding of x, there are at least $m - 2c$ h_j's with $e_1^+ \in E_j(x)$. That is, there are at most $2c$ h_j's with $e_1^+ \notin E_j(x)$. Let H_r be the set of the (at most $2c$) h_j's with $e_1^+ \in E_j(x)$. We use the following method to select the remaining (at most) $2r - 1$ indices.

1. **for** a remaining edge e_g^+ on the ring R **do**
2. **if** there are more than $\frac{c}{r}$ hyperedges $h_j \in H_r$ with $e_g^+ \in E_j(x)$, **then** we select the index q and set $H_r = H_r - \{j \mid e_g^+ \in E_j(x)\}$.
3. **if** the size of H_r is more than $\frac{c}{r}$ **then** goto Step 1 **else** stop.

The above procedure will stop after at most $2r-1$ iterations since each time the size of H_r is reduced by at least $\frac{c}{r}$ and the original size of H_r is at most $2c$.

Now, consider any edge e_i^+ or e_i^- with index i ($1 \leqslant i \leqslant n$) not selected in the above procedure. For the embedding x, the number of h_j's that are not cut at edge e_i^+ or e_i^- in x is at most the size of H_r, that is upper bounded by $\frac{c}{r}$. Thus, the lemma holds. □

Note that, in the proof of Lemma 1, we assume that x is known when selecting the $2r-1$ indices. In fact, x will be the optimal solution that we do not known. However, we can go through all possible sets of $2r-1$ indices in polynomial time. Based on Lemma 1, we can solve the problem as follows:

1. try all possible choices of $i_2, i_3, \ldots, i_{2r}$.
2. for each $h_j \in H$, try the $2r-1$ choices for cutting the path P_k^j containing $e_{i_1}^+, e_{i_2}^+, \ldots,$ or $e_{i_{2r}}^+$.

Step 1 takes $\mathrm{O}(n^{2r-1})$ time and Step 2 needs $\mathrm{O}((2r)^m) = \mathrm{O}((2r)^{\mathrm{O}(\log n)}) = n^{\mathrm{O}(\log 2r)}$ time.

Theorem 2. *There is a PTAS with ratio $1+\frac{1}{r}$ that runs in $\mathrm{O}(n^{2r-1} \times n^{\mathrm{O}(\log 2r)})$ time when $m = \mathrm{O}(\log n)$.*

4. The algorithm for $c \geqslant \mathrm{O}(\log n)$ and $c = \mathrm{O}(m)$

In this section, we consider the case where $c = \mathrm{O}(m)$. We use linear programming and randomized rounding approach. Let $h_j = (u_j, S_j)$ be a hyperedge. We define k_j+1 variables, $x_{j,1}, x_{j,2}, \ldots, x_{j,k_j+1}$. $x_{j,l} = 1$ indicates that P_l^j is cut for the embedding of h_j. For each segment P_q^j of h_j and an edge e_i^+ on the ring, we have a constant $\mu_{i,q,j}$. $\mu_{i,q,j} = 1$ if edge e_i^+ is in the segment P_q^j of h_j. Otherwise, $\mu_{i,q,j} = 0$. We have the following LP formulation.

$$\min c;$$

$$\sum_{l=1}^{k_j+1} x_{j,l} = 1;$$

$$c(e_i^+) = \sum_{j=1}^{m} \sum_{q=1}^{k_j+1} \mu_{i,q,j}(x_{j,q+1} + x_{j,q+2} + \cdots + x_{j,k_j}) \leqslant c; \tag{1}$$

$$c(e_i^-) = \sum_{j=1}^{m} \sum_{q=1}^{k_j+1} \mu_{i,q,j}(x_{j,0} + x_{j,1} + \cdots + x_{j,q-1}) \leqslant c. \tag{2}$$

For a fixed hyperedge h_j and a directed edge e_i^+ (e_i^-) on the ring, there is only one $\mu_{i,q,j}$ for $q = 0, 1, \ldots, k_j$ with value 1. Consider such a segment P_q^j, where e_i^+ and e_i^- are in the segment. e_i^+ is used in the embedding of h_j if one of the segments $P_{q+1}^j, P_{q+2}^j, \ldots, P_{k_j}^j$ is cut for the embedding of h_j. Likewise, e_i^- is used in the embedding of h_j if one of the segments $P_0^j, P_1^j, \ldots, P_{q-1}^j$ is cut for the embedding of h_j. Therefore, we have (1) and (2).

The fractional version of the linear programming problem can be solved in polynomial time. After we get a fractional solution $x'_{j,l}$, independently with probability $x'_{j,l}$, we set $\hat{x}_{j,l} = 1$ and $\hat{x}_{j,h} = 0$ for the rest of h. Thus, we obtain an integer solution $\hat{x}$ for the LP problem. Let c_{opt} be the optimal congestion of the LP formulation. Similar to Lemma 3 in [8], we can prove that

Theorem 3. *Assume that $m \geqslant c_1 \log n$, and $c_{\mathrm{opt}} = c_2 \times m$. Let $\hat{x}$ be the 0–1 solution obtained by randomized rounding. With probability at least $1 - n^{1-\frac{1}{3}\varepsilon^2 c_2^2 c_1}$, for any e_i^+ and e_i^- in E_R,*

$$e_i^+(\hat{x}) \leqslant (1+\varepsilon)c_{\mathrm{opt}},$$

and

$$e_i^-(\hat{x}) \leqslant (1+\varepsilon)c_{\mathrm{opt}}.$$

The proof is given in Appendix A. Using the standard derandomization method for packing integer programs [7], we can have a polynomial deterministic algorithm.

Theorem 4. *There is a PTAS for EDHR when $m \geqslant \mathrm{O}(\log n)$ and $c_{\mathrm{opt}} \geqslant \mathrm{O}(m)$.*

5. The general algorithm

The linear programming and randomized rounding approach in Section 4 does not work for the case where c_{opt} is small comparing with m. Here we propose a method that decomposes the set of all hyperedges into two groups so that we can give approximate embeddings using different methods for the two groups.

Consider $2r$ indices $i_1, i_2, \ldots, i_{2r}$ of edges in E_R. Let $e_{i_1}^+, e_{i_2}^+, \ldots, e_{i_{2r}}^+$ be $2r$ edges on the ring. We define

$$R_{i_1,i_2,\ldots,i_{2r}} = \bigl\{1 \leqslant j \leqslant m \mid \text{there exist an } l \text{ such that } e_{i_k}^+ \in P_l^j \text{ for any } k \in \{1,2,\ldots,2r\}\bigr\}.$$

Let $U_{i_1,i_2,\ldots,i_{2r}} = \{1,2,\ldots,m\} - R_{i_1,i_2,\ldots,i_{2r}}$. Let x_{opt} be an optimal embedding. $x_{\mathrm{opt}}|R_{i_1,i_2,\ldots,i_{2r}}$ and $x_{\mathrm{opt}}|U_{i_1,\ldots,i_{2r}}$

denote the reduced embeddings of x_{opt} on the sets of hyperedges $R_{i_1,i_2,\ldots,i_{2r}}$ and $U_{i_1,i_2,\ldots,i_{2r}}$, respectively.

Lemma 5. *$|U_{i_1,i_2,\ldots,i_{2r}}| \leqslant 4rc_{\text{opt}}$ and $|R_{i_1,i_2,\ldots,i_{2r}}| \geqslant m - 4rc_{\text{opt}}$.*

Proof. Consider an $E_j(x_{\text{opt}})$ containing all the $2r$ edges $e^+_{i_1}, e^+_{i_2}, \ldots, e^+_{i_{2r}}$. For each edge $e^+_{i_k}$, there are at most $2c_{\text{opt}}$ H_j's such that $e^+_{i_k} \notin E_j(x_{\text{opt}})$. Thus, there are at most $4rc_{\text{opt}}$ h_j's in total with $e^+_{i_k} \notin E_j(x_{\text{opt}})$ for some i_k. Therefore, $|R_{i_1,i_2,\ldots,i_{2r}}| \geqslant m - 4rc_{\text{opt}}$.

By definition, $|U_{i_1,i_2,\ldots,i_{2r}}| \leqslant 4rc_{\text{opt}}$. □

Let x^{i_1} be an embedding of h_j's in $R_{i_1,i_2,\ldots,i_{2r}}$ such that every h_j is cut at edge $e^+_{i_1}$. Now, we want to show that

Lemma 6. *For any fixed index $1 \leqslant i_1 \leqslant n$, there exist $2r-1$ indices $1 \leqslant i_2, i_3, \ldots, i_{2r} \leqslant n$ such that for every edge e^+_i and e^-_i in E_R,*

$$e^+_i(x^{i_1}) - e^+_i(x_{\text{opt}}|R_{i_1,i_2,\ldots,i_{2r}}) \leqslant \frac{1}{r}c_{\text{opt}}, \quad \textit{and}$$
$$e^-_i(x^{i_1}) - e^-_i(x_{\text{opt}}|R_{i_1,i_2,\ldots,i_{2r}}) \leqslant \frac{1}{r}c_{\text{opt}}.$$

Proof. To show the existence of the $2r-1$ indices, we give a way to find the $2r-1$ indices assuming that x_{opt} is known. First, we select an arbitrary edge, say, $e^+_1 = (1,2) = (i_1, i_1+1)$ on the ring. In the embedding of x_{opt}, there are at least $m - 2c_{\text{opt}}$ h_j's with $e^+_1 \in E_j(x_{\text{opt}})$. That is, there are at most $2c_{\text{opt}}$ h_j's with $e^+_1 \notin E_j(x_{\text{opt}})$. (If we cut all the m h_j's in H at edge e^+_1, there are at most $2c_{\text{opt}}$ h_j's that are embedded in a way different from that of x_{opt}.) Let H_r be the set of the (at most $2c_{\text{opt}}$) h_j's with $e^+_1 \in E_j(x_{\text{opt}})$.

For every edge e^+_g on the ring, if there are more than c_{opt}/r hyperedges in H_r with $e^+_g \in E_j(x_{\text{opt}})$, then we select the index q. Consider the set $R_{i_1,g}$ of indices. $j \in R_{i_1,g}$ if $e^+_g \in E_j(x_{\text{opt}})$ and $e^+_1 \in E_j(x_{\text{opt}})$. If we cut all the (at most $m - 2c_{\text{opt}}$) h_j's with j in $R_{i_1,g}$ on edge e^+_1, there are at most $2c_{\text{opt}} - c_{\text{opt}}/r$ h_j's that are embedded in a way different from that of x_{opt}. Set $H_r = H_r - \{j \mid e^+_g \in E_j(x_{\text{opt}})\}$ (the set of at most $2c_{\text{opt}} - c_{\text{opt}}/r$ h_j's that are embedded in a way different from that of x_{opt}). If the size of H_r is more than c_{opt}/r then we can repeat the process and find another edge e_g (index). The process continues until the size of H_r is less than c_{opt}/r. The above procedure will stop after at most $2r-1$ iterations since each time the size of H_r is reduced by at least c_{opt}/r and the original size of H_r is at most $2c_{\text{opt}}$.

Now, consider any edge e^+_i with index i $(1 \leqslant i \leqslant n)$ not selected in the above procedure. The number of h_j's in $R_{i_1,i_2,\ldots,i_{2r}}$ that are not cut correctly at edge e^+_i in x^{i_1} is at most the size of H_r, that is upper bounded by c_{opt}/r. Thus, the lemma holds. □

Theorem 7. *There is a PTAS for the EDHR problem.*

Proof. We first compute $U_{i_1,i_2,\ldots,i_r}$ and $R_{i_1,i_2,\ldots,i_{2r}}$.

Case 1. $|U_{i_1,i_2,\ldots,i_{2r}}| \leqslant C\log n$. We use the enumerating approach in Section 3 to compute an embedding for the set of hyperedges in $U_{i_1,i_2,\ldots,i_{2r}}$. For the hyperedges in $R_{i_1,i_2,\ldots,i_{2r}}$, we simply cut the ring at edge e^+_1. From Lemma 6 and Theorem 2, the ratio is $\frac{1}{r} + \frac{1}{r}$.

Case 2. $|U_{i_1,i_2,\ldots,i_{2r}}| > C\log n$. We use the LP and randomized rounding approach in Section 4 to compute an embedding for the set of hyperedges in $U_{i_1,i_2,\ldots,i_{2r}}$. For the hyperedges in $R_{i_1,i_2,\ldots,i_{2r}}$, we simply cut the ring at edge e^+_1. The LP formulation is as follows:

$$\min c;$$
$$\sum_{l=1}^{k_j+1} x_{j,l} = 1 \quad \text{for } j = 1, 2, \ldots, |U_{1_1,i_2,\ldots,i_{2r}}|;$$
$$\sum_{j=1}^{|U_{i_1,i_2,\ldots,i_{2r}}|} \sum_{q=1}^{k_j+1} \mu_{i,q,j}(x_{j,q+1} + x_{j,q+2} + \cdots + x_{j,k_j}) \leqslant c - c(e^+_i|R);$$
$$\sum_{j=1}^{|U_{i_1,i_2,\ldots,i_{2r}}|} \sum_{q=1}^{k_j+1} \mu_{i,q,j}(x_{j,0} + x_{j,1} + \cdots + x_{j,q-1}) \leqslant c - c(e^-_i|R),$$

where $c(e^+_i|R)$ and $c(e^-_i|R)$ are the number of times that e^+_i and e^-_i are used for the embedding of h_j's in $R_{i_1,i_2,\ldots,i_{2r}}$.

Theorem 3 and Lemma 6 ensure that the ratio is $1+\varepsilon$ for any ε. The standard derandomization approach gives a deterministic algorithm. □

Remark. The NP-hardness of the directed version is still open.

Acknowledgement

This work is fully supported by a grant from City University of Hong Kong [Project No. 7001696].

Appendix A. The proof of Theorem 3

Proof. To prove the theorem, we need the following lemma originally from [7].

Lemma 8. *Let $X_1, X_2, \ldots, X_n$ be n independent random 0–1 variables, where x_i takes 1 with probability p_i, $0 < p_i < 1$. Let $X = \sum_{i=1}^{n} X_i$, and $\mu = E[X]$. Then for $\delta > 0$, $\Pr(X > \mu + \delta n) < \exp(-\frac{1}{3}n\delta^2)$.*

For a fixed i and a fixed j, only one $\mu_{i,q,j}$ is 1 and the rest are 0. For $l = 0, 1, \ldots, k_j$, consider such an l with $\mu_{i,q,l} = 1$. For a fixed j, only one $x_{j,l}$ is rounded to 1. Thus, $\mu_{i,q,j}(x_{j,q+1} + x_{j,q+2} + \cdots + x_{j,k_j})$ and $\mu_{i,q,j}(x_{j,0} + x_{j,1} + \cdots + x_{j,q-1})$ are also randomly rounded to either 1 or 0 and are independently for different j's. Therefore, both

$$c(e_i^+) = \sum_{j=1}^{m} \sum_{q=1}^{k_j+1} \mu_{i,q,j}(x_{j,q+1} + x_{j,q+2} + \cdots + x_{j,k_j})$$

and

$$c(e_i^-) = \sum_{j=1}^{m} \sum_{q=1}^{k_j+1} \mu_{i,q,j}(x_{j,0} + x_{j,1} + \cdots + x_{j,q-1})$$

are sums of m independent 0–1 random variables. Set

$$E\big[c(e_i^+)\big] = \sum_{j=1}^{m} \sum_{q=1}^{k_j+1} \mu_{i,q,j} E[x_{j,q+1} + x_{j,q+2} + \cdots + x_{j,k_j}] = \mu_i^+ \leqslant c_{\text{opt}},$$

and

$$E\big[c(e_i^-)\big] = \sum_{j=1}^{m} \sum_{q=1}^{k_j+1} \mu_{i,q,j} E[x_{j,0} + x_{j,1} + \cdots + x_{j,q-1}] = \mu_i^- \leqslant c_{\text{opt}}.$$

From Lemma 8, for any fixed δ,

$$\Pr\big(c(e_i^+) > \mu_i^+ + \delta m\big) \leqslant \exp\left(-\frac{1}{3}\delta^2 m\right).$$

Consider the set of all clockwise edges $E_R^+ = \{e_1^+, e_2^+, \ldots, e_n^+\}$,

$$\Pr\big(c(e_i^+) > \mu_i^+ + \delta m \text{ for at least one } e_i^+ \in E_R\big) \leqslant n \times \exp\left(-\frac{1}{3}\delta^2 m\right).$$

Similarly, we can show that

$$\Pr\big(c(e_i^-) > \mu_i^- + \delta m \text{ for at least one } e_i^- \in E_R\big) \leqslant n \times \exp\left(-\frac{1}{3}\delta^2 m\right).$$

By assumption, $m \geqslant C \log n$. Thus, we have

$$n \times \exp\left(-\frac{1}{3}\delta^2 m\right) \leqslant n^{1-\delta^2 C/3}.$$

Therefore, we get a randomized algorithm to find a solution x for the problem with probability at lease $1 - 2n^{1-\delta^2 C/3}$ such that for any $e_i^+ \in E_R$ and $e_i^- \in E_R$, $c(e_i^+) \leqslant \mu_i + \delta m \leqslant c_{\text{opt}} + \varepsilon c_{\text{opt}}$, and $c(e_i^-) \leqslant c_{\text{opt}} + \varepsilon c_{\text{opt}}$, where $\varepsilon = \frac{\delta}{c}$. □

References

[1] A. Frank, T. Nishizeki, N. Saito, H. Suzuki, E. Tardos, Algorithms for routing around a rectangle, Discrete Appl. Math. 40 (1992) 363–378.

[2] J.L. Ganley, J.P. Cohoon, Minimum-congestion hypergraph embedding in a cycle, IEEE Trans. Comput. 46 (5) (1997) 600–602.

[3] T. Gonzalez, Improved approximation algorithm for embedding hyperedges in a cycle, Inform. Process. Lett. 67 (1998) 267–271.

[4] Q.P. Gu, Y. Wang, Efficient algorithm for embedding hypergraph in a cycle, in: Proc. 10th Internat. Conf. on High Performance Computing, Hyderabad, India, December 2003, pp. 85–94.

[5] S.L. Lee, H.J. Ho, Algorithms and complexity for weighted hypergraph embedding in a cycle, in: Proc. 1st Internat. Symp. on Cyber World (CW2002), 2002.

[6] M. Li, B. Ma, L. Wang, On the closest string and substring problems, J. ACM 49 (2002) 157–171.

[7] R. Motwani, P. Raghavan, Randomized Algorithms, Cambridge Univ. Press, Cambridge, 1995.

[8] X. Deng, G. Li, A PTAS for embedding hypergraph in a cycle (extended abstract), in: Proc. 31st Internat. Colloquium on Automata, Languages and Programming (ICALP 2004), Turku, Finland, 2004, pp. 433–444.

[9] B.S. Baker, R.Y. Pinter, An algorithm for the optimal placement and routing of a circuit within a ring of pads, in: Proc. 24th Symp. Foundations of Computer Science, 1983, pp. 360–370.