Splittable Single Source-Sink Routing on CMP Grids: A Sublinear Number of Paths Suffice

Adrian Kosowski, Przemyslaw Uznanski

Splittable Single Source-Sink Routing on CMP Grids: A Sublinear Number of Paths Suffice

Adrian Kosowski Przemysław Uznański

INRIA Bordeaux – Sud-Ouest, 33405 Talence CEDEX, France
{adrian.kosowski, przemyslaw.uznanski}@inria.fr

Abstract

In single chip multiprocessors (CMP) with grid topologies, a significant part of power consumption is attributed to communications between the cores of the grid. We investigate the problem of routing communications between CMP cores using shortest paths, in a model in which the power cost associated with activating a communication link at a transmission speed of f bytes/second is proportional to f^α, for some constant exponent $\alpha > 2$.

Our main result is a trade-off showing how the power required for communication in CMP grids depends on the ability to split communication requests between a given pair of node, routing each such request along multiple paths. For a pair of cores in a $m \times n$ grid, the number of available communication paths between them grows exponentially with n, m. By contrast, we show that optimal power consumption (up to constant factors) can be achieved by splitting each communication request into k paths, starting from a threshold value of $k = \Theta(n^{1/(\alpha-1)})$. This threshold is much smaller than n for typical values of $\alpha \approx 3$, and may be considered practically feasible for use in routing schemes on the grid. More generally, we provide efficient algorithms for routing multiple k-splittable communication requests between two cores in the grid, providing solutions within a constant approximation of the optimum cost. We support our results with algorithm simulations, showing that for practical instances, our approach using k-splittable requests leads to a power cost close to that of the optimal solution with arbitrarily splittable requests, starting from the stated threshold value of k.

1 Introduction

The increase in the level of integration of single chip multiprocessors (CMPs) creates demand for high-speed communication on-chip, which in turn increases the power consumption on CMP. This trend is predicted to continue in the future [7]. Numerous studies concern the optimization of power cost in integrated chip designs, taking into account that both processors and communication buses may operate at variable frequency, determining the speed of computations or transmissions (cf. [8, 12, 14, 16]). The increase of power cost with the third power of workload in such designs is a well-established relation (cf. e.g. [3, 6, 16]).

A significant part of power in CMPs is consumed by maintaining communications within the chip, and that makes efficient allocation of communication routes a very important issue [15]. On CMP grids, links with dynamic frequency and/or voltage scaling are used ([13, 17]), and the dissipated power P on a link is related to the frequency f and voltage V on it by the following relation supported by both theory and experiments $P \sim f \cdot V^2$ (cf. e.g. [2]). However, for most designs, an increase in operating frequency also results in an increase in voltage, roughly according to the relation $V \sim f$ (cf. e.g. [17]), which results in the relation between power cost and transmission speed given as $P \sim f^3$ ([4, 5]). Such a model of power consumption was recently studied in the context of splittable Manhattan-path routing by Benoit *et al.* [4]. They introduced several routing schemes in an effort to minimize the total power cost, but observed that this may require the splitting of each communication request, and routing its fragments along a potentially very large number of communication paths. Splitting a request, taking care of the route for each part, and merging it at the target imposes additional time and power overhead.

In this work, our goal in this work is to show how to *limit path splitting as much as possible*, without excessively increasing communication power cost. Specifically, we consider the problem of optimizing the power consumption cost of communication between two given cores, which may sometimes require the routing of multiple requests. (Our scenario can also be seen as a rough approximation of the general case of multi-core communication, under the simplifying assumption that the total communication rate due to communication between all pairs of cores other than the distinguished pair may be treated as the same for each link, and so excluded from optimization.) Our power consumption model assumes that if an edge is transmitting at rate f, the power cost of maintaining the frequency over an edge is proportional to f^{α} for a given constant $\alpha > 2$, identical for every edge. We make the practically-motivated assumption [13, 17] that only the dynamic part (associated with transmission) is dominant for high communication rate, and static effects need not be considered in optimization.

Outline and results. Our study concerns routing between a single source-sink pair of nodes using Manhattan paths on a grid CMP. Communication between these nodes is assumed to be static, i.e. constant over time, and the cost of a transmission along an edge is assumed to be proportional to a fixed power of the transmission rate. The considered model, power cost function, and rules of routing are formally presented in Section 2. We briefly outline the theory of Manhattan-path routing with arbitrarily splittable requests (Max-MP). We provide an optimal convex programming formulation of the problem, leading to a routing scheme denoted as $\text{O}\small{\text{PT}}$, and recall the properties of the $\mathcal{C}$ routing scheme introduced in [4]. We also provide a convenient formulation of Manhattan routings in terms of transmission through nodes.

Our main results are given in Section 3. They concern the variant of the Manhattan routing problem in which each request can be satisfied by at most k communication paths, where k is a parameter of the model (k-MP). We study the value of the ratio of the cost of the optimal solution in this case, denoted $\text{O}\small{\text{PT}}_k$, to the cost of the routing scheme $\text{O}\small{\text{PT}}$ with arbitrarily splittable paths. We establish that in general, $\text{cost}(\text{O}\small{\text{PT}}_k)/\text{cost}(\text{O}\small{\text{PT}}) = O(1+\frac{n}{k^{\alpha-1}})$, whereas for the special case of $d \geq 1$ identical requests of the same size, this ratio is given precisely as $\Theta(1+\frac{n}{(kd)^{\alpha-1}})$. This means that for $k = o(n^{1/(\alpha-1)})$, the requirement that requests can be split into at most k paths impacts the cost of the routing scheme asymptotically, i.e., increases the cost by an unbounded factor for sufficiently large n. On the other hand, for k larger than the threshold value of $\Theta(n^{1/(\alpha-1)})$, the obtained k-splittable routings are within a constant factor of the optimal solution to Max-MP.

The proposed bounds are obtained through the analysis of three efficiently implementable algorithmic schemes for solving k-MP: $\mathcal{F}_k$ routing and $\mathcal{D}_k$ routing (for uniform requests), and $\mathcal{A}_k$ (for non-uniform requests). The latter two are shown to have a constant approximation ratio with respect to the cost of $\text{O}\small{\text{PT}}_k$ for all k, while the former converges to the cost of $\text{O}\small{\text{PT}}$ as k goes to infinity. The design of such approximate techniques results from the observation that $\text{O}\small{\text{PT}}_k$ is NP-hard.

Finally, in Section 4, we perform a validation, using simulations, of the determined threshold value of $k = \Theta(n^{1/(\alpha-1)})$, showing the effect of smaller and larger values of k on the cost of the routing. We also experimentally compare the performance on $\mathcal{F}_k$ routing and $\mathcal{D}_k$ routing, studying their convergence to asymptotic behavior for increasing values of k and different values of the power cost exponent $\alpha \approx 3$.

2 Framework

Platform and power consumption model. We model our platform as a grid graph on a set of $m \times n$ uniform nodes $V_{i,j}$, with $1 \leq i \leq m$ and $1 \leq j \leq n$. Without loss of generality, we assume that $m \geq n$. We will also assume for the purpose of analysis that the sides of the grid are of the same order of magnitude, i.e., $m = O(n)$. Nodes are connected by bidirectional edges. The horizontal edge $E_{i,j}$ connects $V_{i,j}$ and

$V_{i,j+1}$ (for $1 \le i \le m$, $1 \le j \le n-1$), and the vertical edge $E'_{i,j}$ connects $V_{i,j}$ and $V_{i+1,j}$ ($1 \le i \le m-1$, $1 \le j \le n$), see Fig. 1 for an illustration.

The power consumed on each edge is closely related to the amount of data sent through this edge in a unit of time. To simplify the analysis of the model, we discard constant factors, and (following [4]) set the cost of transmission at rate x as $C(x) = x^{\alpha}$, where $\alpha > 2$ is an absolute constant of the model (it is reasonable to assume $\alpha \approx 3$).

Communication and routing rules. The study of routing with Manhattan-type paths (of shortest length) is motivated by practical concerns, in particular, the need to minimize communication latency, and to confine communications between nearby processors to a local area of the grid. For the purpose of the study of single source-sink communications, it is assumed that the source and target are placed in the opposite corners of the grid; for communications between a different pair of nodes, considerations can be restricted to the respective rectangular sub-grid.

A *routing* R of a single communication request of size s is a weighted set of paths, $\{(w_1, p_1), \dots, (w_k, p_k)\}$, where each path p_i starts at the same source vertex $V_{1,1}$, and ends at the same target vertex $V_{m,n}$ in the opposite corner of the grid. The real-valued weights w_i satisfy $w_i \ge 0$ and $\sum_i w_i = s$. This definition of a routing extends naturally to a set of $d \ge 1$ requests, which may be *uniform* (with identical request size $s = K/d$), or *non-uniform* (with possibly distinct request sizes $s_1, \dots, s_d$). Given a routing R, we define $R(e)$ as the size of the transmission going through an edge e, i.e.: $R(e) = \sum_{i:\ e \in p_i} w_i$. (We will use this notation accordingly for routings denoted by letters different from R.)

The *routing policy* is expressed by the bound k on the splitability of each request:

- In *k-Path Manhattan Routing (k-*MP*)*, communication for each request can be split into any number of $k' \le k$ (partially overlapping) source-sink paths, where k is a parameter of the model.
- In *Max-Paths Manhattan Routing (*Max-MP*)*, the number of paths allowed for each request is unbounded ($k = +\infty$).

Problem definition. For a given routing policy with parameter k and power coefficient α, we define our optimization problem as follows: *Given a $m \times n$ grid and a set of requests of sizes $(s_1, \dots, s_d)$, with $\sum_{i=1}^{d} s_i = K$, find a routing R of this set of requests minimizing the total power cost of transmission through all the edges of the grid, expressed by the cost function:*

$$\mathrm{cost}(R) = \sum_{i=1}^{m} \sum_{j=1}^{n-1} R(E_{i,j})^{\alpha} + \sum_{i=1}^{m-1} \sum_{j=1}^{n} R\left(E'_{i,j}\right)^{\alpha}.$$

Solution to Max-MP **Routing.** For Max-MP, the routing policy does not impose a bound on k. We will denote the optimal solution to Max-MP by OPT and use it as a reference for k-splittable routing algorithms. The adopted definition of routing cost leads directly to a convex-programming formulation of Max-MP routing, and thus applications of convex programming algorithms lead to polynomial-time schemes with arbitrarily good approximation of OPT (cf. e.g. [1, 10] for a discussion of convex programming in the context of finding min-cost flows).

We remark on the following lower bound on the size of OPT. Consider any Max-MP routing which transmits requests of total size K. The edges adjacent to node $V_{1,1}$, i.e., $\{E_{1,1}, E'_{1,1}\}$, have to transmit requests of size K in total. It follows that:

$$\mathrm{cost}(\mathrm{OPT}) \ge \left(\tfrac{K}{2}\right)^{\alpha} = \Theta(K^{\alpha}). \tag{1}$$

Remarkably, as shown in [4], this lower bound is tight regardless of the size of the grid, since it can be achieved using a specific routing scheme. We will provide a definition of a scheme called $\mathcal{C}$ which has equivalent properties, but is described from a different perspective, based on load balancing on so-called *vertex diagonals*. We will then use this scheme as a starting point for schemes solving k-MP.

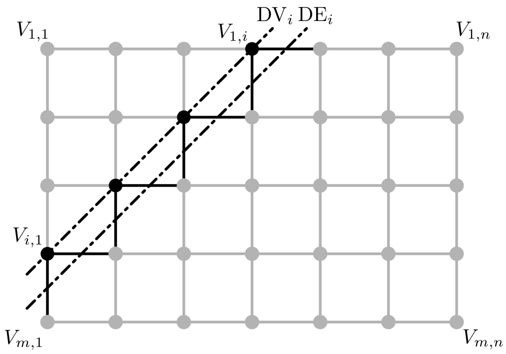

Figure 1: Vertex diagonal DV_i and edge diagonal DE_i

Our approach: load balancing on vertex diagonals. In all of the routing schemes which we propose in this paper, we will attempt to perform "load balancing" of paths with respect to transmission through vertices rather than edges. Hence, in a similar fashion to the notation $R(e)$ for an edge e, we define $R(v)$ as the total transmission size going through a vertex v in routing R.

We introduce the notion of the l-th *vertex diagonal*, denoted as DV_l ($1 \leq l \leq n+m-1$) by splitting the set of vertices according to their distance from the source, as follows (see Fig. 1 for an illustration): $V_{i,j} \in \mathrm{DV}_l$, iff $i+j = l+1$. Likewise, by the l-th *edge diagonal*, denoted DE_l ($1 \leq l \leq n+m-2$), we mean the set of edges connecting vertices from DV_l and DV_{l+1}, namely: $E_{i,j}, E'_{i,j} \in \mathrm{DE}_l$, iff $i+j = l+1$.

We start by observing that the values of $R(v)$ uniquely determine the values of $R(e)$. This property will allow us to design routing schemes simply by setting $R(v)$ for all nodes.

Routing scheme $\mathcal{C}$ for Max-MP. We define the routing scheme $\mathcal{C}$ for Max-MP by putting a limit on the transmission going through vertices. Since each diagonal of vertices has a total transmission of exactly K, we set an equal value of transmission for all vertices in the layer:

$$\forall_{v \in \mathrm{DV}_j} \mathcal{C}(v) = \frac{K}{|\mathrm{DV}_j|}. \tag{2}$$

To verify that this routing is well-defined, we compute transfers over each edge based on the transfers on vertices (a complete implementation and analysis is provided in the Appendix[1]). Examples of transfers obtained using this algorithm are shown in Fig. 2.

The scheme $\mathcal{C}$ corresponds to the differently formulated algorithm studied in [4], where it was shown that it admits a constant approximation ratio for Max-MP.

Theorem 2.1 ([4])**.** $\mathrm{cost}(\mathcal{C}) = \Theta(K^\alpha) = \Theta(\mathrm{cost}(\textsc{Opt}))$.

Although such a solution has optimal, up to a constant factor, power cost, it can result in a single request being split into a large number of paths. Indeed, for a given graph $G = (V, E)$ and any flow f on G, f can be represented as the union of at most $|E|$ weighted paths. It follows that both OPT routing (computed through convex optimization) and $\mathcal{C}$ routing require $O(nm)$ splits per request. In the next section, we will show that it is possible to preserve a constant approximation ratio of the optimal cost, while using a much smaller number of splits, sublinear in the dimensions of the grid.

3 Schemes for k-Splittable Routing

In this section, we present three schemes for solving the k-Path Manhattan Routing problem (k-MP). The first two, denoted $\mathcal{F}_k$ and $\mathcal{D}_k$, are designed for uniform sets of requests. As the bound k on the

[1] A full version of the paper is also available at: http://hal.inria.fr/hal-00737611

number of allowed paths per request tends to infinity, these approaches will be shown to converge to the performance of schemes OPT and $\mathcal{C}$ for Max-MP, respectively. The third scheme, denoted $\mathcal{A}_k$, is an extension of $\mathcal{D}_k$ which also works for non-uniform sets of requests.

3.1 1-splittable routing with uniform requests

We start by considering the 1-MP routing policy, meaning that no splitting of requests is allowed. We can treat this problem as a discrete version of a continuous Max-MP problem. First, we will consider uniform requests (of equal sizes); without loss of generality, we can assume that the input consists of d requests of size 1, each.

This considered problem can be solved by the flow-based $\mathcal{F}_1$ routing approach presented in Algorithm 1. The obtained solution is optimal, i.e., for uniform instances, we have $\text{cost}(\text{OPT}_1) = \text{cost}(\mathcal{F}_1)$. Moreover, using a classical min-cost flow algorithm, a $\mathcal{F}_1$ routing can be found in polynomial time with

Algorithm 1 $\mathcal{F}_1$ routing scheme {optimal solution to uniform 1-MP}

Input: A set of d unsplittable requests of size $s = 1$ in a $m \times n$ grid.
Solution:

1. Construct a multigraph G' such that $V(G') = V(G)$.
2. For every directed edge $e \in E(G)$, add d weighted directed edges to G', having the same endpoints as e, and weights given as: $1^\alpha, 2^\alpha - 1^\alpha, \ldots, d^\alpha - (d-1)^\alpha$.
3. Return the min-cost flow of size d in G', using the two opposite corners of the grid as the source and sink.

respect to parameters n, m, and d.

We will now provide asymptotic bounds on the size of the (optimal) solution to the uniform 1-MP problem. We obtain the lower bound by combining the lower bound for problem Max-MP (formula (1) with $K = d$), with an additional factor resulting from the discrete nature of 1-MP.

Lemma 3.1. *For every $R \in$ uniform 1-MP: $\text{cost}(R) = \Omega(d^\alpha) + \Omega(nd)$.*

(Proofs omitted due to space constraints are provided in the Appendix.)

To provide a complementary upper bound on the size of 1-MP routings, we do not analyze the optimal scheme $\mathcal{F}_1$, but instead propose an approximation scheme called $\mathcal{D}_1$ routing, which turns out to be easier to analyze.

We design the $\mathcal{D}_1$ routing through a discretization of the construction of $\mathcal{C}$ routing proposed in the previous section for Max-MP. Similarly to equation (2), we will place limits on the size of the transfer going through vertices. Consider the vertex diagonal DV_p with $1 \le p \le n+m-1$, and let $i = |\text{DV}_p|$. Suppose that the vertices of DV_p are ordered by decreasing first coordinate, as $\text{DV}_p = \{v_1, \ldots, v_i\}$. Then, for $1 \le j \le i$, we successively set $\mathcal{D}_1(v_j)$ so that at each step, the following condition holds: $\mathcal{D}_1(v_1) + \ldots + \mathcal{D}_1(v_j) = \lfloor d \cdot \frac{j}{i} \rfloor$. This is achieved by setting:

$$\mathcal{D}_1(v_j) = \left\lfloor d \cdot \tfrac{j}{i} \right\rfloor - \left\lfloor d \cdot \tfrac{j-1}{i} \right\rfloor. \tag{3}$$

To verify the correctness of this construction, we deduce transfer values over vertical and horizontal edges from values over vertices; a formal implementation of $\mathcal{D}_1$ routing is provided in Algorithm 2. An exemplary comparison of the vertex and edge transfers for $\mathcal{C}$ routing and $\mathcal{D}_1$ routing is shown in Fig. 2.

We start the analysis of the cost of $\mathcal{D}_1$ routing with the following lemma.

Lemma 3.2. *Let* DV *be an arbitrary vertex diagonal, and let* $|\text{DV}| = i$. *Then:*

$$\sum_{v \in DV} \mathcal{D}_1(v)^\alpha = \begin{cases} i\left(\left(\frac{d}{i}\right)^\alpha + O\left(\left(\frac{d}{i}\right)^{\alpha-2}\right)\right), & \text{for } i < d \\ d, & \text{for } i \ge d. \end{cases}$$

Algorithm 2 $\mathcal{D}_1$ routing scheme {for uniform 1-MP}

Input: A set of d unsplittable requests of size $s = 1$ in a $m \times n$ grid.
Solution: For each diagonal DE_j of the grid, $1 \leq j < n+m$, set the flow on its successive horizontal edges e_i and vertical edges e'_i, $1 \leq i \leq j$, as follows:

- If $1 \leq j < n$, set:

$$\mathcal{D}_1(e_i) = \left\lfloor d\tfrac{i}{j} \right\rfloor - \left\lfloor d\tfrac{i}{j+1} \right\rfloor, \qquad \mathcal{D}_1(e'_i) = \left\lfloor d\tfrac{i}{j+1} \right\rfloor - \left\lfloor d\tfrac{i-1}{j} \right\rfloor.$$

- If $n \leq j < m$, set:

$$\mathcal{D}_1(e_i) = \left\lfloor d\tfrac{i}{n} \right\rfloor - \left\lfloor d\tfrac{i-1}{n} \right\rfloor, \qquad \mathcal{D}_1(e'_i) = 0.$$

- If $m \leq j < n+m$, set:

$$\mathcal{D}_1(e_i) = \left\lfloor d\tfrac{i}{n+m-j} \right\rfloor - \left\lfloor d\tfrac{i-1}{n+m-j-1} \right\rfloor, \qquad \mathcal{D}_1(e'_i) = \left\lfloor d\tfrac{i}{n+m-j-1} \right\rfloor - \left\lfloor d\tfrac{i}{n+m-j} \right\rfloor.$$

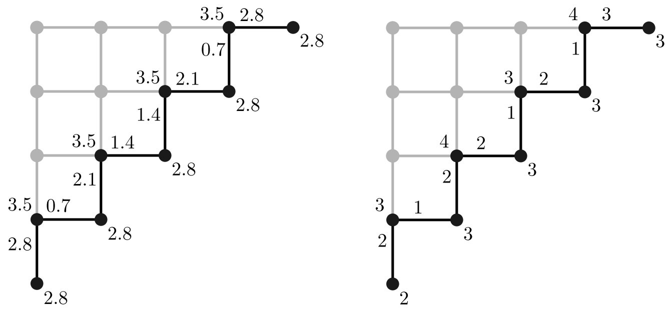

Figure 2: Comparison of transfer values over one diagonal for a $\mathcal{C}$ routing with $K = 14$ (on the left) and a $\mathcal{D}_1$ routing with $d = 14$ (on the right)

Using the above lemma, we compute the cost of a $\mathcal{D}_1$ routing as $\mathrm{cost}(\mathcal{D}_1) = \Theta(d^\alpha) + \Theta(nd)$. By Lemma 3.1, this cost is asymptotically the best possible for 1-MP.

Theorem 3.3. *For a uniform set of d requests (with total size $K = d$):*

$$\mathrm{cost}(\mathcal{D}_1) = \Theta(d^\alpha) + \Theta(nd) = \mathrm{cost}(\mathcal{F}_1).$$

3.2 k-splittable routing with uniform requests

We now proceed to extend our results from the previous section to the case of k-MP uniform routing. We will consider sets of d requests of total size K, i.e., of size K/d each. A natural generalization of $\mathcal{D}_1$ routing, called $\mathcal{D}_k$ routing, is presented in Algorithm 3.

Algorithm 3 $\mathcal{D}_k$ routing scheme {for uniform k-MP}

Input: A set of d k-splittable requests, of size K/d each, in a $m \times n$ grid.
Solution: Split each of the requests into k smaller ones, each of size $\frac{K}{kd}$. Return the $\mathcal{D}_1$ routing of this new set of requests.

Since in a $\mathcal{D}_k$ routing, we split the transmission of each request equally along its k paths, the cost of such a routing is the same as that of a $\mathcal{D}_1$ routing on the extended set of kd requests of size $\frac{K}{kd}$ each. Hence, the following result follows directly from Theorem 3.3 by a scaling argument: $\mathrm{cost}(\mathcal{D}_k) = \Theta(K^\alpha) + \Theta(K^\alpha \frac{n}{(kd)^{\alpha-1}})$. Next, we show that although $\mathcal{D}_k$ only splits requests into paths of equal weight, one cannot achieve a better asymptotic result by using unequal splits, i.e., for any $R \in$ uniform k-MP : $\text{cost}(R) = \Omega(K^\alpha) + \Omega(K^\alpha \frac{n}{(kd)^{\alpha-1}})$. Combining these results, we obtain the following theorem, stating the optimality of $\mathcal{D}_k$ in the class of k-splittable routings.

Theorem 3.4. *For a uniform set of d requests with total size K:* $\text{cost}(\mathcal{D}_k) = \Theta(K^\alpha) + \Theta\left(K^\alpha \frac{n}{(kd)^{\alpha-1}}\right) = \text{cost}(\text{OPT}_k)$, *where* OPT_k *denotes the optimal cost solution to the considered set of requests for* k-MP.

Combining the bound on $\text{cost}(\text{OPT}_k)$ in the above Theorem with the bound on $\text{cost}(\text{OPT})$ in Theorem 2.1 for Max-MP routing, we obtain our main result: the threshold value of k for which imposing a limit of k into which each request can be split does not affect the asymptotics of power cost.

Theorem 3.5. *For uniform requests, imposing a routing policy with a split limit of* $k = \Theta\left(\frac{1}{d} \cdot n^{\frac{1}{\alpha-1}}\right)$ *does not affect the power cost, i.e.:* $\text{cost}(\text{OPT}_k) = \Theta(\text{cost}(\text{OPT}))$.

We end this subsection with a remark on the asymptotic behavior of the considered routing schemes for uniform instances, when $k \to +\infty$. Taking into account Theorems 2.1 and 3.4, we obtain:

Proposition 3.6. *For a grid of fixed dimension:* $\lim_{k\to+\infty} \text{cost}(\mathcal{D}_k) = \text{cost}(\mathcal{C})$.

Since, in general $\text{cost}(\mathcal{C}) > \text{cost}(\text{OPT})$, it is natural to ask for a different routing schemes for k-MP with improved limit behavior. A natural candidate is $\mathcal{F}_k$ routing, obtained by a natural generalization of $\mathcal{F}_1$ routing, as given by Algorithm 4.

Algorithm 4 $\mathcal{F}_k$ routing scheme {for uniform k-MP}

Input: A set of d k-splittable requests, of size K/d each, in a $m \times n$ grid.
Solution: Split each of the requests into k smaller ones, each of size $\frac{K}{kd}$. Return the $\mathcal{F}_1$ routing of the new set of requests.

This algorithm turns out to by asymptotically optimal as $k \to +\infty$.

Proposition 3.7. *For a grid of fixed dimension:* $\lim_{k\to+\infty} \text{cost}(\mathcal{F}_k) = \text{cost}(\text{OPT})$.

3.3 k-splittable routing with non-uniform requests

We close our considerations with a discussion of the general (non-uniform) case, where no assumptions are made about the sizes of the routed requests. We first observe that the considered problem is computationally hard.

Theorem 3.8. *The following decision version of non-uniform* 1-MP *routing is NP-complete: "Given* $(n, m, K = (K_1, \ldots, K_i), C, \alpha)$*, decide if it is possible to perform 1-MP routing with cost* $\leq C$*."*

Despite the hardness of the studied problem, one can try to look for approximate solutions. Note that applying $\mathcal{D}_k$ routing naively to a set of non-uniform requests could lead to excessively large additional cost. However, by applying a careful modification of $\mathcal{D}_k$ routing, called the $\mathcal{A}_k$ routing scheme (Algorithm 5), we obtain a good tool for routing non-uniform requests on the grid.

Theorem 3.9. *For non-uniform requests,* $\mathcal{A}_k$ *finds a solution to* k-MP *whose cost is within a constant factor of the optimum k-splittable routing:* $\text{cost}(\mathcal{A}_k) = \Theta(\text{cost}(\text{OPT}_k))$.

We end this section with a similar threshold theorem as Theorem 3.5 for the uniform case, obtaining bounds on value of k for which a split limit of k no longer affects the asymptotics of the cost of the routing. However, in this case the threshold depends on the structure of the set of requests, hence we only provide lower and upper bounds.

Algorithm 5 $\mathcal{A}_k$ routing scheme {for non-uniform k-MP}

Input: A set of d k-splittable requests, of given sizes $S = (s_1, s_2, \dots, s_d)$ (with $1 = s_1 \leq s_2 \leq \dots \leq s_d$), in a $m \times n$ grid.

Solution:

1. Partition the set of request sizes into the union of disjoint subsets, $S = S_0 \cup S_1 \cup \dots$, such that $\forall_{s \in S_i}\ 2^i \leq s < 2^{i+1}$.
2. For all non-empty sets S_i:
   - Find a $\mathcal{D}_k$ routing for the uniform instance consisting of $|S_i|$ requests of size 2^{i+1} each.
   - For all $1 \leq j \leq |S_i|$, route the j-th input request belonging to S_i using the paths assigned to the j-th request in the corresponding $\mathcal{D}_k$ routing.

Theorem 3.10. *For non-uniform requests, imposing a routing policy with a split limit of k:*

1. *does not affect the asymptotic power cost (i.e.* $\mathrm{cost}(\textsc{Opt}_k) = \Theta(\mathrm{cost}(\textsc{Opt}))$*) when* $k = \Omega\left(n^{\frac{1}{\alpha-1}}\right)$,
2. *always increases the asymptotic power cost (i.e.* $\mathrm{cost}(\textsc{Opt}_k) = \omega(\mathrm{cost}(\textsc{Opt}))$*) when* $k = o\left(\frac{1}{d} \cdot n^{\frac{1}{\alpha-1}}\right)$.

4 Simulations results

In this section we provide the results of experimental evaluation, through simulations, of the algorithms presented in the previous section. We analyze the effect of n, k and α on the efficiency of solutions found for k-MP routing of instances with uniform (identical-size) requests. Throughout the section, we choose the number of requests as $d = 1$ (for uniform instances, other values result only in a scaling factor for k in k-MP, and do not affect Max-MP).

We focus on the approximation ratio, looking at the cost of the routing obtained using the two schemes designed for uniform k-MP ($\mathcal{D}_k$, $\mathcal{F}_k$), relative to the cost of the optimal solution $\textsc{Opt}$ to Max-MP, which is treated as the reference solution. In some graphs, we also provide the cost of the sub-optimal Max-MP routing $\mathcal{C}$ as an additional reference. Keep in mind, that both $\mathcal{C}$ and $\textsc{Opt}$ use as much as $\Theta(nm)$ routing paths.

We recall that the value of cost of the optimal solution to Max-MP, cost($\textsc{Opt}_k$), is bounded from below by cost($\textsc{Opt}$), and from above by both cost($\mathcal{D}_k$) and cost($\mathcal{F}_k$).

The implementation and tests were performed within a software package, written by the authors for this purpose in GNU C++. The min-cost flow subroutines were implemented using the standard cycle-canceling method [11] with some optimizations for faster performance. The presented results of the tests are deterministic and fully reproducible, independent of the test environment and the details of the implementation of the flow algorithms.

Impact of k on the routing cost. We begin by studying the approximation ratio of algorithms $\mathcal{F}_k$ and $\mathcal{D}_k$ for increasing values of k, the allowed number of splits of each requests. In the first plot (Fig. 3), we fix the dimensions of the grid $n, m = 30$, model power cost exponent $\alpha = 2.5$, plotting the values of $\mathrm{cost}(\mathcal{F}_k)/\mathrm{cost}(\textsc{Opt})$ and $\mathrm{cost}(\mathcal{D}_k)/\mathrm{cost}(\textsc{Opt})$ for k in the range $k \in [10, 100]$. For reference, we also provide the approximation ratio of $\mathcal{C}$ routing for the studied instance.

We observe that, as predicted by theory (Propositions 3.6 and 3.7), $\lim_{k\to+\infty} \mathrm{cost}(\mathcal{F}_k) = \mathrm{cost}(\textsc{Opt})$ and $\lim_{k\to+\infty} \mathrm{cost}(\mathcal{D}_k) = \mathrm{cost}(\mathcal{C})$, and the respective costs converge to their limits quickly, reaching a point 10% over the respective limit already for $k < n$. In general, the convergence need not be monotone for either of the approximation algorithms, since partitioning a request into $k+1$ equally-weighted paths may give worse results than partitioning it into k equally-weighted paths.

In our second plot (Fig. 4), we present more compelling evidence of the relation $k = \Theta\left(n^{1/(\alpha-1)}\right)$ for the threshold split value resulting in asymptotically-optimal cost, derived theoretically as Theorem 3.5.

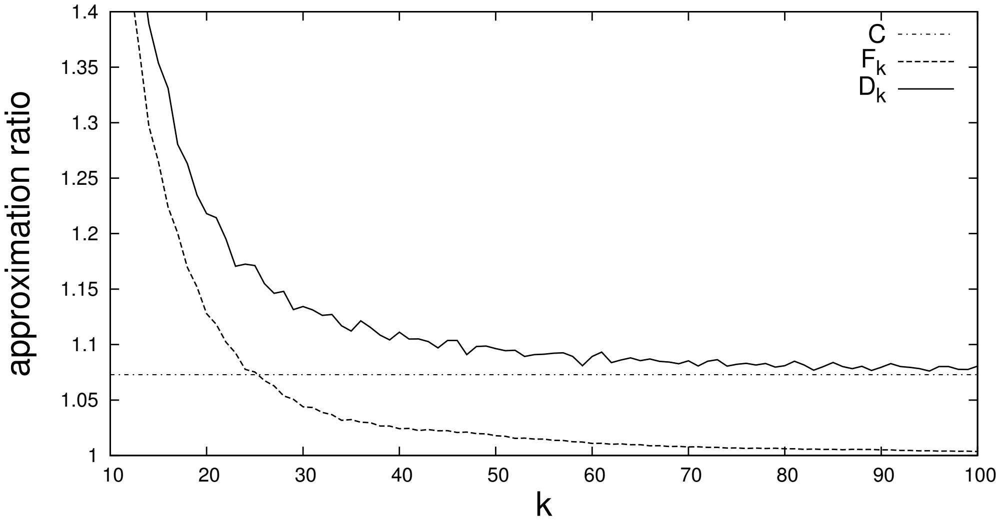

Figure 3: Effect of k on the cost of $\mathcal{F}_k$ and $\mathcal{D}_k$ routing for a 30×30 grid

Once again, we choose model parameter $\alpha = 2.5$. In the experiment, we consider square grids of increasing size in the range $n = m \in [10, 120]$, testing three different relations between n and k ($k = \lfloor 2n^{1/2} \rfloor$, $k = \lfloor \frac{3}{2} n^{2/3} \rfloor$, $k = n$). For each of these relations, we plot the approximation ratios $\text{cost}(\mathcal{F}_k)/\text{cost}(\text{OPT})$ and $\text{cost}(\mathcal{D}_k)/\text{cost}(\text{OPT})$. Based on the plot, we can presume that:

- For the relation $k \sim n^{1/2}$, we have in the limit:

$$\text{cost}(\mathcal{F}_k)/\text{cost}(\text{OPT}) \to +\infty, \qquad \text{cost}(\mathcal{D}_k)/\text{cost}(\text{OPT}) \to +\infty.$$

- For the relation $k \sim n^{2/3}$, we have in the limit:

$$\text{cost}(\mathcal{F}_k)/\text{cost}(\text{OPT}) \to \mathit{const}, \qquad \text{cost}(\mathcal{D}_k)/\text{cost}(\text{OPT}) \to \mathit{const}.$$

- For the relation $k \sim n$, we have in the limit:

$$\text{cost}(\mathcal{F}_k)/\text{cost}(\text{OPT}) \to 1, \qquad \text{cost}(\mathcal{D}_k)/\text{cost}(\text{OPT}) \to \mathit{const}.$$

We remark that the relation $k \sim n^{2/3}$ precisely corresponds to the threshold exponent $1/(\alpha - 1) = 2/3$ for the considered value of α. Thus, the limit behavior of all the algorithms is consistent with the theory derived in the previous section. We note that the cost achieved by both $\mathcal{F}_k$ routing and $\mathcal{D}_k$ routing is highly satisfactory, and that the performance of $\mathcal{F}_k$ routing proves to be superior to $\mathcal{D}_k$ in all of the performed tests.

Effect of power exponent α. In auxiliary experiments, we studied the effect of the power exponent α (which is a constant of the model) on the required threshold value of split parameter k. We tested the rate of convergence of the approximation ratio $\text{cost}(\mathcal{F}_k)/\text{cost}(\text{OPT})$ to 1 in a grid of dimensions $n = m = 30$ for three different values of the power exponent, $\alpha \in \{2.5, 3, 3.5\}$. It was observed that the convergence is faster for larger values of α. This is consistent with the theoretical threshold, $k = \Theta\left(n^{1/(\alpha-1)}\right)$, whose growth rate decreases with the increase of α.

5 Conclusions

The contribution of our study is twofold. On the one hand, we advance the theory of splitting of requests in Manhattan routing on the grid, and point out that in practice, only a relatively small number of splits per request will be beneficial from a power-cost perspective. On the other hand, we propose

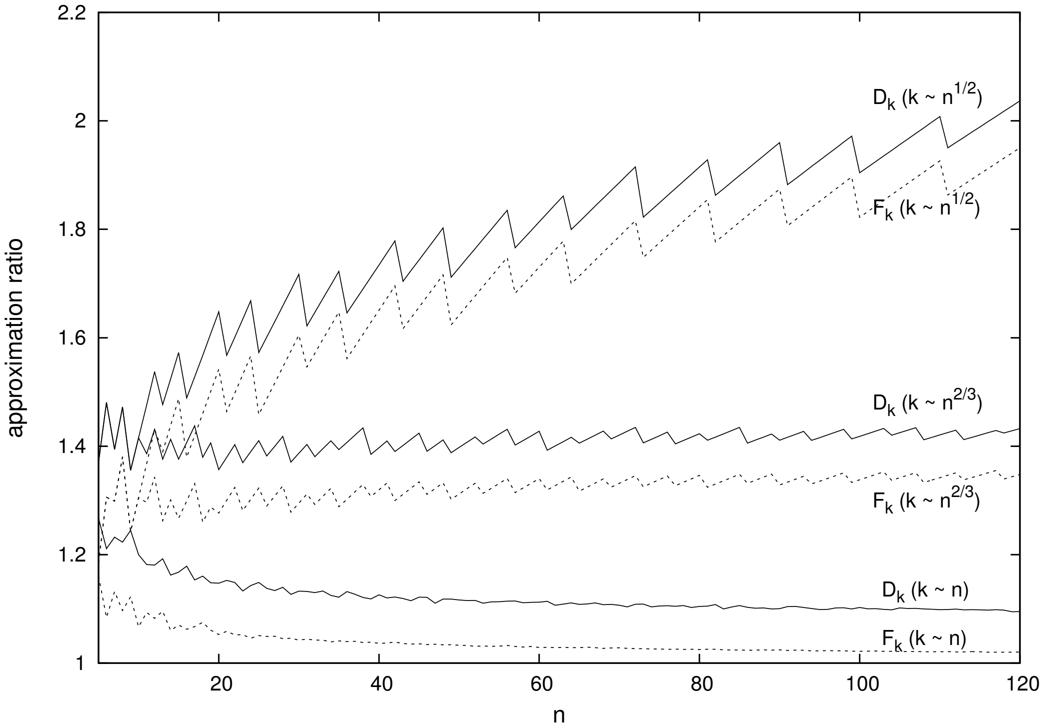

Figure 4: Approximation ratio for $\mathcal{F}_k$ and $\mathcal{D}_k$ routing with split parameter $k \sim n^\beta$, for β greater, equal and smaller than $1/(\alpha-1)$. $[\alpha = 2.5]$

efficient approximation schemes for such a k-path routing problem. Simulations provide evidence that corroborates the theoretical results, showing that the designed algorithms lead to routings with a cost which is, in practice, even superior to that resulting from our theoretical bounds.

In future work, it would be beneficial to improve the constant bounds on the approximation ratios of our algorithms, establishing more tightly their dependence on the power exponent α. Another promising direction of study would extend our results to routing requests between multiple sources and targets on the grid. Such a study would have a purely experimental nature, since the thresholds which appear in multi-core communication scenarios are difficult to capture theoretically, depending on the observed traffic patterns.

References

[1] R. K. Ahuja, D. S. Hochbaum, and J. B. Orlin. Solving the convex cost integer dual network flow problem. In *Proc. Int. Conf. on Integer Programming and Combinatorial Optimization*, volume 1610 of *LNCS*, pp. 31–44, 1999.

[2] A. Andrei, M. T. Schmitz, P. Eles, Z. Peng, and B. M. Al-Hashimi. Simultaneous communication and processor voltage scaling for dynamic and leakage energy reduction in time-constrained systems. In *Proc. Int. Conf. on Computer-Aided Design*, pp. 361–367. IEEE, 2004.

[3] H. Aydin and Q. Yang. Energy-aware partitioning for multiprocessor real-time systems. In *Proc. Int. Parallel and Distributed Processing Symp.*, page 113. IEEE, 2003.

[4] A. Benoit, R. G. Melhem, P. Renaud-Goud, and Y. Robert. Power-aware manhattan routing on chip multiprocessors. In *Proc. Int. Parallel and Distributed Processing Symp.*, pp. 189–200. IEEE, 2012.

[5] A. Benoit, P. Renaud-Goud, and Y. Robert. On the performance of greedy algorithms for power consumption minimization. In *Proc. Int. Conf. on Parallel Processing*, pp. 454–463. IEEE, 2011.

[6] J.-J. Chen and T.-W. Kuo. Multiprocessor energy-efficient scheduling for real-time tasks with different power characteristics. In *Proc. Int. Conf. on Parallel Processing*, pp. 13–20. IEEE, 2005.

[7] R. D. G. Blake and T. Mudge. A survey of multicore processors. *Signal Processing Magazine*, pp. 26(6):26–37, 2009.

[8] T. Ishihara and H. Yasuura. Voltage scheduling problem for dynamically variable voltage processors. In *Proc. Int. Symp. on Low Power Electronics and Design*, pp. 197–202, 1998.

[9] N. Karmarker and R. M. Karp. The differencing method of set partitioning. Technical Report UCB/CSD-83-113, EECS Department, University of California, Berkeley, 1983.

[10] A. V. Karzanov and S. T. McCormick. Polynomial methods for separable convex optimization in unimodular linear spaces with applications. *SIAM J. on Computing*, 26(4):1245–1275, 1997.

[11] M. Klein. A primal method for minimal cost flows. *Management Sci.*, 14:205–220, 1967.

[12] P. Langen and B. Juurlink. Leakage-aware multiprocessor scheduling. *J. of Signal Processing Systems*, 57:73–88, 2009.

[13] S. E. Lee and N. Bagherzadeh. A variable frequency link for a power-aware network-on-chip (NoC). *Integration*, 42(4):479–485, 2009.

[14] R. Mishra, N. Rastogi, D. Zhu, D. Mossé, and R. Melhem. Energy aware scheduling for distributed real-time systems. In *Proc. Int. Parallel and Distributed Processing Symp.*, page 21, 2003.

[15] J. D. Owens, W. J. Dally, R. Ho, D. J. Jayasimha, S. W. Keckler, and L.-S. Peh. Research challenges for on-chip interconnection networks. *IEEE Micro*, 27:96–108, 2007.

[16] K. Pruhs, R. van Stee, and P. Uthaisombut. Speed scaling of tasks with precedence constraints. In *Proc. Int. Workshop on Approximation and Online Algorithms*, volume 3879 of *LNCS*, pp. 307–319, 2005.

[17] L. Shang, L.-S. Peh, and N. K. Jha. Dynamic voltage scaling with links for power optimization of interconnection networks. In *Proc. Int. Symp. on High-Performance Computer Architecture*, pp. 91–102. IEEE, 2003.

Appendix: Omitted proofs

Proof of Lemma 3.1. The bound $\Omega(d^\alpha)$ holds since we cannot get a better solution than the one for Max-MP with the same set of requests. Since each message of size 1 induces a cost at least $n+m-2$, by the convexity of the cost function, we get a total cost of $\Omega(nd)$ for d of them. $\square$

In the subsequent proofs we use the following Lemma.

Lemma 5.1. *For arbitrary $R \in MP$, we can compute the values of $R(e)$ from values of $R(v)$.*

Proof. We will look at DE_j for various values of j. We denote $\mathrm{DV}_j = \{v_1, v_2, \dots\}$ and $\mathrm{DV}_{j+1} = \{u_1, u_2, \dots\}$, with vertices reverse-ordered by first coordinate. Also, the edges in $\mathrm{DE}_j = \{e_1, \dots\} \cup \{e'_1, \dots\}$ are assumed to be ordered respectively horizontal and vertical edges.

1. Let $1 \le j < n$, and $1 \le i \le j$.

$$R(e_i) = (R(v_1) + \ldots + R(v_i)) - (R(u_1) + \ldots + R(u_i))$$

$$R(e'_i) = (R(u_1) + \ldots + R(u_i)) - (R(v_1) + \ldots + R(v_{i-1}))$$

2. For $n \le j < m$, we can perform similar reasoning:

$$R(e_i) = (R(v_1) + \ldots + R(v_i)) - (R(u_1) + \ldots + R(u_{i-1}))$$
$$R(e'_i) = (R(u_1) + \ldots + R(u_i)) - (R(v_1) + \ldots + R(v_i))$$

3. For $m \le j < n+m$:

$$R(e_i) = (R(v_1) + \ldots + R(v_i)) - (R(u_1) + \ldots + R(u_{i-1}))$$
$$R(e'_i) = (R(u_1) + \ldots + R(u_i)) - (R(v_1) + \ldots + R(v_i))$$

$\square$

Proof of Lemma 3.2. By equation (3) we have (substituting $d' = d \bmod i$):

$$\sum_{v \in \mathrm{DV}} \mathcal{D}_1(v)^\alpha = \sum_{j=1}^{i} \left(\left\lfloor d \cdot \frac{j}{i} \right\rfloor - \left\lfloor d \cdot \frac{j-1}{i} \right\rfloor \right)^\alpha = \left\lfloor \frac{d}{i} \right\rfloor^\alpha \cdot (i - d') + \left(\left\lfloor \frac{d}{i} \right\rfloor + 1 \right)^\alpha \cdot d'$$

We consider two cases:

- ($i < d$) Substituting $x = \frac{d}{i}$, $\lambda = \frac{d}{i} - \left\lfloor \frac{d}{i} \right\rfloor = \frac{d'}{i}$:

$$\sum_{v \in \mathrm{DV}} \mathcal{D}_1(v)^\alpha = (x-\lambda)^\alpha (1-\lambda) i + (x+1-\lambda)^\alpha \lambda i =$$

(using Taylor's series theorem, for some $x - \lambda \le u_1 \le x \le u_2 \le x + 1 - \lambda$)

$$= \left(x^\alpha - \lambda \alpha x^{\alpha-1} + \frac{\lambda^2}{2} \alpha (\alpha-1) u_1^{\alpha-2} \right) (1-\lambda) i +$$

$$+ \left(x^\alpha + (1-\lambda) \alpha x^{\alpha-1} + \frac{(1-\lambda)^2}{2} \alpha (\alpha-1) u_2^{\alpha-2} \right) \lambda i \le$$

(since $u_1 \le x$ and $u_2 \le x + 1 - \lambda \le 2x$)

$$\le i x^\alpha + i \alpha (\alpha-1) \left(\frac{\lambda^2}{2} (1-\lambda) + \frac{(1-\lambda)^2}{2} \lambda 2^\alpha \right) x^{\alpha-2} =$$

$$= i \left(x^\alpha + O\left(x^{\alpha-2}\right) \right) = i \left(\left(\frac{d}{i} \right)^\alpha + O\left(\left(\frac{d}{i} \right)^{\alpha-2} \right) \right).$$

- ($i \geq d$) The sum simplifies to:

$$\sum_{v \in \mathrm{DV}} \mathcal{D}_1(v)^\alpha = 0 \cdot (i - (d \bmod i)) + 1 \cdot (d \bmod i) = d.$$

$\square$

Proof of Theorem 3.3. The lower bound follows from Lemma 3.1. We observe that $\mathrm{cost}(\mathcal{F}_1) \leq \mathrm{cost}(\mathcal{D}_1)$. Now, to put upper bound on cost of the $\mathcal{D}_1$ routing scheme, we will consider two cases:

- ($d \geq n$) Then, using Lemma 3.2 and inequalities (4-5), we have:

$$\mathrm{cost}(\mathcal{D}_1) = \sum_{i=1}^{n+m-2} \sum_{e \in \mathrm{DE}_i} \mathcal{D}_1(e)^\alpha = \sum_{i=n}^{m-1} \sum_{e \in \mathrm{DE}_i} \mathcal{D}_1(e)^\alpha + 2 \sum_{i=1}^{n-1} \sum_{e \in \mathrm{DE}_i} \mathcal{D}_1(e)^\alpha \leq$$

$$\leq (m-n) \sum_{v \in \mathrm{DV}_n} \mathcal{D}_1(v)^\alpha + 2 \sum_{i=2}^{n} \sum_{v \in \mathrm{DV}_i} \mathcal{D}_1(v)^\alpha =$$

$$= (m-n) n \left(\left(\frac{d}{n} \right)^\alpha + O\left(\left(\frac{d}{n} \right)^{\alpha-2} \right) \right) + 2 \sum_{i=2}^{n} i \left(\left(\frac{d}{i} \right)^\alpha + O\left(\left(\frac{d}{i} \right)^{\alpha-2} \right) \right).$$

However, we have:

$$(m-n) \cdot n \cdot \left(\frac{d}{n} \right)^\alpha + 2 \sum_{i=2}^{n} i \left(\frac{d}{i} \right)^\alpha = (m-n) \frac{d^\alpha}{n^{\alpha-1}} + 2 \sum_{i=2}^{n} \frac{d^\alpha}{i^{\alpha-1}} \leq$$

$$\leq (m-n) \frac{d^\alpha}{n^{\alpha-1}} + 2 \int_1^n \frac{d^\alpha}{x^{\alpha-1}} dx = d^\alpha \cdot \left(\frac{m-n}{n} \frac{1}{n^{\alpha-2}} + \frac{2}{(\alpha-2)} \left(1 - \frac{1}{n^{\alpha-2}} \right) \right) = \Theta(d^\alpha).$$

It follows that:

$$\mathrm{cost}(\mathcal{D}_1) \leq \Theta(d^\alpha) + (m-n) n \cdot O\left(\left(\frac{d}{n} \right)^{\alpha-2} \right) + O\left(\sum_{i=2}^{n} i \left(\frac{d}{i} \right)^{\alpha-2} \right) =$$

$$= \Theta(d^\alpha) + O\left(d^{\alpha-2} n^{4-\alpha} \right) + O\left(n \cdot \max\left(d^{\alpha-2}, d^{\alpha-2} n^{3-\alpha} \right) \right) =$$

$$= \Theta(d^\alpha) + O\left(d^{\alpha-2} n^{4-\alpha} \right) + O\left(d^{\alpha-2} n \right) + O\left(d^{\alpha-2} n^{4-\alpha} \right) = \Theta(d^\alpha),$$

where the last step holds since $1 \leq n \leq d$ and $\alpha > 2$.

- ($d < n$) In this case, we have:

$$\mathrm{cost}(\mathcal{D}_1) = \sum_{i=1}^{n+m-2} \sum_{e \in \mathrm{DE}_i} \mathcal{D}_1(e)^\alpha = \sum_{i=d}^{n+m-d+1} \sum_{e \in \mathrm{DE}_i} \mathcal{D}_1(e)^\alpha + 2 \sum_{i=1}^{d-1} \sum_{e \in \mathrm{DE}_i} \mathcal{D}_1(e)^\alpha \leq$$

$$\leq (n+m-2d+2) d + 2 \sum_{i=2}^{d} \sum_{v \in DV_i} \mathcal{D}_1(v)^\alpha = O(nd) + 2 \sum_{i=2}^{d} i \left(\left(\frac{d}{i} \right)^\alpha + O\left(\left(\frac{d}{i} \right)^{\alpha-2} \right) \right) =$$

$$= O(nd) + \Theta(d^\alpha) + O\left(d \max\left(d^{\alpha-2}, d \right) \right) = O(nd) + \Theta(d^\alpha).$$

$\square$

We remark on the close connections between the costs measured in terms of nodes and edges. For any Manhattan routing R we have:

$$R(E_{i,j-1}) + R\left(E'_{i-1,j}\right) = R(V_{i,j}) = R(E_{i,j}) + R\left(E'_{i,j}\right)$$

Since the cost function $f(x) = x^\alpha$ is convex, we get:

$$R(V_{i,j})^\alpha \geq R(E_{i,j})^\alpha + R\left(E'_{i,j}\right)^\alpha$$

which leads to useful inequalities between the costs related to vertices and edges:

$$\sum_{v \in \mathrm{DV}_l} R(v)^{\alpha} \geq \sum_{e \in DE_l} R(e)^{\alpha} \tag{4}$$

$$\sum_{v \in \mathrm{DV}_l} R(v)^{\alpha} \geq \sum_{e \in DE_{l-1}} R(e)^{\alpha}. \tag{5}$$

Proof of Theorem 3.4. We have already established that for Max-MP, cost(R) is in $\Omega(K^{\alpha})$, which also holds for k-MP. For the second part of the lower bound, we use convexity of the cost function and the fact that on any diagonal, the stream of the routing of the d requests can be split into at most kd distinct paths:

$$\sum_{e \in \mathrm{DE}} R(e)^{\alpha} \geq (kd) \left(\frac{K}{kd} \right)^{\alpha}$$

Thus:

$$\mathrm{cost}(R) = \sum_{i=1}^{n+m-2} \sum_{e \in \mathrm{DE}_i} R(e)^{\alpha} \geq (n+m-2)(kd) \left(\frac{K}{kd} \right)^{\alpha} = \Theta\left(K^{\alpha} \frac{n}{(kd)^{\alpha-1}} \right).$$

□

Proof of Theorem 3.5. To prove sufficiency, we set $k = \Omega(\frac{1}{d} \cdot n^{\frac{1}{\alpha-1}})$ and by Theorem 3.4 we obtain:

$$\mathrm{cost}(\textsc{Opt}_k) = \Theta(K^{\alpha}) + \Theta\left(K^{\alpha} \frac{n}{(kd)^{\alpha-1}} \right) = \Theta(K^{\alpha}) + o(K^{\alpha}) = \Theta(K^{\alpha}).$$

To prove necessity, we observe that with $k = o(\frac{1}{d} \cdot n^{\frac{1}{\alpha-1}})$, the lower bound on cost becomes:

$$\mathrm{cost}(\textsc{Opt}_k) = \Omega(K^{\alpha}) + \Omega\left(K^{\alpha} \frac{n}{o(n)} \right) = \omega(K^{\alpha}).$$

□

Proof of Proposition 3.6. Denote by $\textsc{Opt}'$ an assignment of transfer sizes to edges of G obtained by rounding up the value of the transfer size in an $\textsc{Opt}$ routing to the nearest integer multiple of $\frac{K}{kd}$, i.e., for all edges e, $\textsc{Opt}'(e) = \frac{K}{kd} \lceil \frac{kd}{K} \textsc{Opt}(e) \rceil$. By the properties of the min-cost flow used in the design of $\mathcal{F}_k$, the assignment $\textsc{Opt}'$ majorizes the cost of the $\mathcal{F}_k$ routing for the considered instance. Thus, we have:

$$\mathrm{cost}(\mathcal{F}_k) \leq \mathrm{cost}(\textsc{Opt}') \leq \mathrm{cost}(\textsc{Opt}) + O\left(n \cdot \frac{K}{kd} \right).$$

Taking the limit $k \to +\infty$, the claim follows. □

Proof of Theorem 3.8. The proof proceeds by reduction from PARTITION PROBLEM [9]: "Given a set S of integers, decide if it is possible to partition S into subsets S_1 and S_2 such that $\sum S_1 = \sum S_2$."

For such an instance, we select the instance of 1-MP routing as follows: $n = 2$, $m = 2$, $K = S$ and $C = 4 \left(\frac{1}{2} \sum_{s \in S} s \right)^{\alpha}$.

Observe that by identifying the sets of requests routed along the 2 different paths of the grid with S_1 and S_2, we have:

$$\mathrm{cost}(S_1, S_2) = 2 \left(\sum_{s \in S_1} s \right)^{\alpha} + 2 \left(\sum_{s \in S_2} s \right)^{\alpha} \geq 4 \left(\frac{1}{2} \sum_{s \in S} s \right)^{\alpha} = C$$

and equality in the bound is achieved only if $\sum_{s \in S_1} s = \sum_{s \in S_2} s$. □

Proof of Theorem 3.9. It is enough to prove that for each edge diagonal DE, the cost induced by $\mathcal{A}_k$ on this diagonal is bounded by a constant with relation to the possible cost of the optimal solution $\textsc{Opt}_k \in k$-MP on this diagonal.
We keep the notation $c_i = |S_i|$, and recall that DV denotes the vertex diagonal adjacent to DE. Also, let $p = |DV|$, and let $\mathcal{D}[S_i]$ refer to the discrete $\mathcal{D}_1$ routing scheme applied inside S_i. We have:

$$\mathrm{cost}\,(\mathrm{DE}, \mathcal{A}_k) = \sum_{e \in \mathrm{DE}} \mathcal{A}_k\,(e)^\alpha \le \sum_{v \in \mathrm{DV}} \mathcal{A}_k\,(v)^\alpha \le$$

(taking into account that for $s \in S_i$ we have $s < 2 \cdot 2^i$)

$$\le \sum_{v \in \mathrm{DV}} \left(\sum_i \mathcal{D}[S_i]\,(v) \cdot \frac{2 \cdot 2^i}{k} \right)^\alpha =$$

$$= 2^\alpha \sum_{v \in \mathrm{DV}} \left(\sum_{i:\ c_i \cdot k \le p} \left(\mathcal{D}[c_i]\,(v) \cdot \frac{2^i}{k} \right) + \sum_{i:\ c_i \cdot k > p} \left(\mathcal{D}[c_i]\,(v) \cdot \frac{2^i}{k} \right) \right)^\alpha =$$

(taking into account that $\forall_{a,b \ge 0}\,(a+b)^\alpha \le 2^{\alpha-1}\,(a^\alpha + b^\alpha)$)

$$\le 2^{2\alpha-1} \sum_{v \in \mathrm{DV}} \left(\left(\sum_{i:\ c_i \cdot k \le p} \mathcal{D}[c_i]\,(v) \cdot \frac{2^i}{k} \right)^\alpha + \left(\sum_{i:\ c_i \cdot k > p} \mathcal{D}[c_i]\,(v) \cdot \frac{2^i}{k} \right)^\alpha \right).$$

Observe that $\mathcal{D}[c_i](v) \le \lceil \frac{c_i \cdot k}{p} \rceil$.
Consequently, for $c_i \cdot k \le p$, we have $\mathcal{D}[c_i](v) \in \{0,1\}$, and since $2^d + 2^{d-1} + \ldots < 2 \cdot 2^d$:

$$\sum_{v \in \mathrm{DV}} \left(\sum_{i:\ c_i \cdot k \le p} \mathcal{D}[c_i]\,(v) \cdot \frac{2^i}{k} \right)^\alpha \le$$

$$\le \sum_{v \in \mathrm{DV}} \left(2 \cdot \max_{i:\ c_i k \le p} \left\{ \mathcal{D}[c_i]\,(v) \cdot \frac{2^i}{k} \right\} \right)^\alpha \le$$

$$\le 2^\alpha \sum_{v \in \mathrm{DV}} \sum_{i:\ c_i \cdot k \le p} \left(\mathcal{D}[c_i]\,(v) \cdot \frac{2^i}{k} \right)^\alpha = 2^\alpha \sum_{i:\ c_i \cdot k \le p} k \cdot c_i \left(\frac{2^i}{k} \right)^\alpha$$

For $c_i k > p$, we have $\mathcal{D}[c_i](v) \le \lceil \frac{c_i \cdot k}{p} \rceil \le 2 \frac{c_i k}{p}$:

$$\sum_{v \in \mathrm{DV}} \left(\sum_{i:\ c_i \cdot k > p} \mathcal{D}[c_i]\,(v) \cdot \frac{2^i}{k} \right)^\alpha \le \sum_{v \in \mathrm{DV}} \left(\sum_{i:\ c_i k > p} 2 \cdot \frac{c_i \cdot k}{p} \cdot \frac{2^i}{k} \right)^\alpha =$$

$$= 2^\alpha \cdot \left(\frac{1}{p} \sum_{i:\ c_i k > p} c_i \cdot 2^i \right)^\alpha \cdot p$$

Adding the two sums, we can write in general:

$$\mathrm{cost}\,(\mathrm{DE}, \mathcal{A}_k) \le 2^{3\alpha-1} \cdot \left(\sum_{i:\ c_i k \le p} c_i \left(\frac{2^i}{k} \right)^\alpha + p \cdot \left(\frac{1}{p} \sum_{i:\ c_i k > p} c_i \cdot 2^i \right)^\alpha \right)$$

Now we proceed to lower-bound the cost induced on diagonal DE by routing $\textsc{Opt}_k$. By $\textsc{Opt}_k[S_i]$, we will denote $\textsc{Opt}_k$ restricted to requests from S_i, and by $\textsc{Opt}_k[s]$ we will understand $\textsc{Opt}_k$ restricted to request s. We have:

$$\mathrm{cost}\,(\mathrm{DE}, \textsc{Opt}_k) = \sum_{e \in \mathrm{DE}} \textsc{Opt}_k\,(e)^\alpha \ge \frac{1}{2^{\alpha-1}} \sum_{v \in \mathrm{DV}} \textsc{Opt}_k\,(v)^\alpha =$$

$$= \frac{1}{2^{\alpha-1}} \sum_{v \in \mathrm{DV}} \left(\sum_{i:\ c_i k \le p} \mathrm{O{\scriptsize PT}}_k[S_i](v) + \sum_{i:\ c_i k > p} \mathrm{O{\scriptsize PT}}_k[S_i](v) \right)^{\alpha} \ge$$

$$\ge \frac{1}{2^{\alpha-1}} \sum_{v \in \mathrm{DV}} \left(\left(\sum_{i:\ c_i k \le p} \mathrm{O{\scriptsize PT}}_k[S_i](v) \right)^{\alpha} + \left(\sum_{i:\ c_i k > p} \mathrm{O{\scriptsize PT}}_k[S_i](v) \right)^{\alpha} \right)$$

We put bounds on both parts of sum:

$$\sum_{v \in \mathrm{DV}} \left(\sum_{i:\ c_i k \le p} \mathrm{O{\scriptsize PT}}_k[S_i](v) \right)^{\alpha} = \sum_{v \in \mathrm{DV}} \left(\sum_{i:\ c_i k \le p} \sum_{s \in S_i} \mathrm{O{\scriptsize PT}}_k[s](v) \right)^{\alpha} \ge$$

$$\ge \sum_{i:\ c_i k \le p} \sum_{s \in S_i} \sum_{v \in \mathrm{DV}} \mathrm{O{\scriptsize PT}}_k[s](v)^{\alpha} \ge$$

(observing that $\sum_{v \in \mathrm{DV}} \mathrm{O{\scriptsize PT}}_k[s](v) = s$, and s can be split into at most k parts)

$$\ge \sum_{i:\ c_i k \le p} \sum_{s \in S_i} k \cdot \left(\frac{s}{k} \right)^{\alpha} \ge \sum_{i:\ c_i k \le p} c_i k \left(\frac{2^i}{k} \right)^{\alpha}$$

In the second part, we obtain:

$$\sum_{v \in \mathrm{DV}} \left(\sum_{i:\ c_i k > p} \mathrm{O{\scriptsize PT}}_k[S_i](v) \right)^{\alpha} \ge$$

$$\ge p \cdot \left(\frac{1}{p} \sum_{v \in \mathrm{DV}} \sum_{i:\ c_i k > p} \mathrm{O{\scriptsize PT}}_k[S_i](v) \right)^{\alpha} =$$

$$= p \cdot \left(\frac{1}{p} \sum_{i:\ c_i k > p} \sum_{s \in S_i} s \right)^{\alpha} \ge p \cdot \left(\frac{1}{p} \sum_{i:\ c_i k > p} c_i \cdot 2^i \right)^{\alpha}$$

Merging both results, we have:

$$\mathrm{cost}(\mathrm{DE}, \mathrm{O{\scriptsize PT}}_k) \ge$$

$$\ge \frac{1}{2^{\alpha-1}} \left(\sum_{i:\ c_i k \le p} c_i k \left(\frac{2^i}{k} \right)^{\alpha} + p \cdot \left(\frac{1}{p} \sum_{i:\ c_i k > p} c_i \cdot 2^i \right)^{\alpha} \right).$$

Combining the lower-bound on the cost of $\mathrm{O{\scriptsize PT}}_k$ and the upper bound on the cost of $\mathcal{A}_k$, we finally have:

$$\mathrm{cost}(\mathrm{DE}, \mathcal{A}_k) \le 2^{4\alpha-2} \mathrm{cost}(\mathrm{DE}, \mathrm{O{\scriptsize PT}}_k),$$

which proves that $\mathcal{A}_k$ is a $(2^{4\alpha-2})$-approximation algorithm for non-uniform k-MP. □

Proof of Theorem 3.10.

1. The optimal-cost routing for a set of d requests of total size K, each of which can be split into k paths, cannot be better than the optimal-cost routing on a single request of size K, which can be split into kd parts. The latter routing problem belongs to uniform $(k \cdot d)$-MP and so, taking into account Theorem 3.4, we can write:

$$\mathrm{cost}(\mathrm{O{\scriptsize PT}}_k) = \Omega\left(K^{\alpha} + K^{\alpha} \frac{n}{(kd)^{\alpha-1}} \right) = \Omega\left(K^{\alpha} + K^{\alpha} \frac{n}{o(n)} \right) = \omega(K^{\alpha}) = \omega(\mathrm{cost}(\mathrm{O{\scriptsize PT}}))$$

2. Only the upper bound needs to be shown. Observe that the cost of an optimal k-MP routing cannot decrease if we replace a pair of requests of size s_1, s_2 by a single request of size $s_1 + s_2$. By iterating the argument, we can upper-bound the value of $\mathrm{O{\scriptsize PT}}_k$ for a given instance of d requests of total size K by the value of $\mathrm{O{\scriptsize PT}}_k$ for an instance consisting of a single request of size K. Once again, the claim follows by an application of Theorem 3.4.

□