Approximations to Maximum Weight Matching Scheduling Algorithms of Low Complexity

Claus Bauer, Dolby Laboratories, San Francisco, USA, cb@dolby.com

Abstract

The choice of the scheduling algorithm is a major design criteria of a switch. Whereas it is known that maximum weight matching algorithms guarantee the stability of an input queued switch, their computational complexity does not allow their practical deployment. In consequence, researchers have designed scheduling algorithms of low complexity and with satisfying performance features.

We extend this field of research by investigating the application of matching algorithms of low complexity that approximate maximum weight matching algorithms as scheduling algorithms for input queued switches. We prove that an algorithm that approximates a maximum weight matching algorithm with approximation parameters (c, d), stabilizes a combined input/output-queued switch with a speed-up of $\frac{1}{c}$. As an application, we show that four known scheduling algorithms of low complexity stabilize a combined input/output-queued switch with a speed-up of two. Finally, we prove that the $improve\_matching$ algorithm can stabilize an input-queued switch when it is deployed with a speed-up of $\frac{3}{2} + \epsilon$.

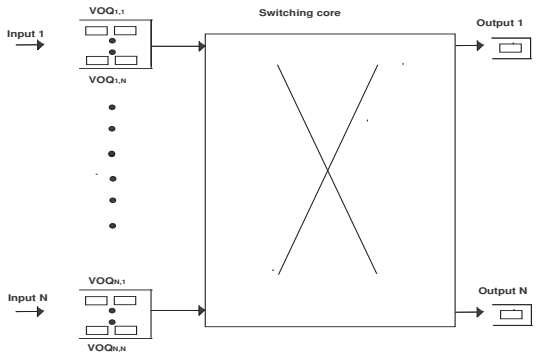
1 Introduction

The architecture of most switches is based on either a pure input (IQ) or a combined input and output (CIOQ) buffered architecture. A typical CIOQ $N \times N$ switch is shown in figure 1. At each input, the arriving cells are buffered into N Virtual Output Queues ($VOQs$). The cells arriving at input i and destined for output j are buffered in $VOQ_{i,j}$, $1 \leq i, j \leq N$. The switching core itself is modeled as a crossbar requiring that not more than one packet can be sent simultaneously from the same input or to the same output. The switching core works at a speed-up $S \geq 1$, which is defined as the ratio between the potentially higher speed of the switching core and the line-speed of the incoming (and outgoing) links.

The scheduling algorithm of a switch determines the stability and delay characteristics of the switch. In the ground breaking work by McKeown [10], the scheduling problem, i.e., the decision which cells shall be forwarded through the swicth and which cells shall remain buffered at the inputs, is modeled as a problem originating from graph theory: the calculation of a maximum weight matching for a bipartite matching. The weights are chosen proportional to the VOQ lengths and and the inputs on one side and the outputs of the switch on the other side constitute the two parts of the bipartite graph. For each input-output pair connected by the matching, a packet is sent from the input to the output. An algorithm based on the Hungarian method solves the maximum weight matching (MWM) problem with a complexity of $O(N^3)$ [9]. It is shown in [10] that the deployment of a maximum weight matching algorithm guarantees the stability of a CIOQ switch without any speed-up requirements, i.e., a speed-up of $S = 1$.

The high complexity of the MWM algorithm shows a need to design faster scheduling algorithms with performance characteristics similar to those of the MWM algorithm. Researchers [1], [4] have focused on maximal weight matching algorithms and have shown that they stabilize a CIOQ switch when they are deployed with a speed-up of 2 or slightly less [2], [3].

The scope of this paper is to design new scheduling algorithms of low complexity with low speed-up requirements. In particular, we investigate the application of matching algorithms of low complexity that approximate



$Fig.1$ $Architecture of an input queued switch$

MWM algorithms with a positive performance ratio as scheduling algorithms for CIOQ switches. Typically [7], a matching algorithm is said to have a performance ratio of c, $0 < c \leq 1$, compared with a MWM algorithm if for all graphs it finds a matching with a weight of at least c times the weight of an optimal solution as computed by a MWM algorithm.

Here, we generalize the concept of the performance ratio as given in [7] as follows. We say that a matching algorithm approximates a MWM algorithms with approximation parameters (c, d), $0 < c \leq 1$, $d \geq 0$, if for any graph the sum of the constant d and of the weight calculated by the matching algorithm is at least as large as c times the optimal weight calculated by the MWM.

In this paper, we first prove general results for matching algorithms that approximate a MWM algorithm with approximation parameters (c, d). We discuss two modes to deploy these algorithms in a CIOQ switch. We show that in either of these modes, a deployment of the switch with a rational speed-up $S \geq \frac{1}{c}$ is sufficient to guarantee the stability of a CIOQ switch.

Then, we apply our results to four known matching algorithms that approximate a MWM algorithm with approximation parameters $(\frac{1}{2}, d)$ and a computational complexity of $O(N^2)$. We show that their deployment with a rational speed-up $S \geq 2$ guarantees the stability of a CIOQ switch.

Finally, we discuss the new $improve\_matching$ algorithm from [7] that approximates a MWM algorithm with approximation parameters $(\frac{2}{3} - \epsilon, 0)$ and a computational complexity of $O(N^2)$. Applying our general results on approximation algorithms, we show that this algorithm stabilizes a CIOQ switch with a rational speedup of $S \geq \frac{3}{2} + \epsilon$.

Thus, whereas previously only maximal matching algorithms that required a speed-up of $S = 2$ were considered as an alternative to the MWM algorithm, here we show for the first time the existence of an algorithm that stabilizes a CIOQ switch with a rational speed-up of $S \geq \frac{3}{2} + \epsilon$.

2 Terminology and Model

Using common terminology from graph theory, we denote a graph G as $G = (V, E)$ where V denotes the set of vertices and E denotes the set of edges of the graph. The letter e always denotes an edge. We assume a weighted graph, where $w(e)$ denotes the weight of the edge e and $w(P)$ denotes the sum of the weights of the edges e belonging to a set of edges $P \subseteq E$. A matching M is a subset of the edges of G such that no two edges in M are incident to the same vertex.

We define $\lambda_{i,j}$ as the expected arrival rate of cells at the $VOQ_{i,j}$. In the sequel we assume that the incoming traffic is *admissible,* i.e., we require that $\max(\max_i \sum_{j=1}^{N} \lambda_{i,j}, \max_j \sum_{i=1}^{N} \lambda_{i,j}) < 1$. We describe the time t via a discrete, slotted time model. Cells are supposed to be of fixed size. An external times lot is the time needed by a cell to arrive completely at an incoming link. As the switching core works at a speedup $S \geq 1$, an internal time slot is defined as the time needed to transfer a cell through the switching core from an input to an output. Throughout this paper we suppose S to be a rational number $S = \frac{b_1}{a_1} \geq 1$, $a_1, b_1 \in \mathbb{N}$, a_1 and b_1 are prime to each other. We suppose a switch to start its operation at time $n = 0$. Then for each positive integer m, the a_1 external time slots $[ma_1, ma_1+1)$, $..[(m+1)a_1-1, (m+1)a_1)$, consist of the b_1 internal time slots $[ma_1, ma_1 + \frac{a_1}{b_1})$, $[ma_1 + \frac{a_1}{b_1}, ma_1 + \frac{2a_1}{1b})$, $... [(m+1)a_1 - \frac{a_1}{b_1}, (m+1)a_1)$. We suppose that cells arrive at the beginning of an external time slot t and are transferred instantly at the end of an internal time slot. We define the norm of an $N \times N$ matrix as $||x|| = \sum_{i,j} x_{i,j}$. The arrival matrix A_t describes the cell arrivals at the $VOQs$:

$$A_t^{i,j} = \left\{ \begin{array}{ll} 1, & \text{if an arrival at } VOQ_{i,j} \text{ occurs at at time } t, \\ 0, & \text{else.} \end{array} \right\}$$

By our model, $A_t^{i,j}$ is always equal to zero if the time t does not correspond to the beginning of an external time slot. The service matrix $D_t(ALGO)$, indicating which queues are served at time t by a scheduling algorithm $ALGO$ is defined as:

$$D_t^{i,j}(ALGO) = \left\{ \begin{array}{ll} 1, & \text{if } VOQ_{i,j} \text{ is served at time } t. \\ 0, & \text{else.} \end{array} \right\}$$

By our model model, $D_t^{i,j}(ALGO)$ is always zero if the time t does not coincide with the end of an internal timeslot. Throughout this paper, we will assume that the weights used by the scheduling algorithms considered are proportional to the lengths of the $VOQs$. We define by $X_t = (X_t^{i,j})_{1 \leq i,j \leq N}$ the occupancy matrix of the $VOQs$ which counts the number of cells buffered in the $VOQs$ at time t. By definition [10], the choice of the matching by a MWM algorithm at the beginning of an (external) timeslot n can be described by the relation

$$D_n(MWM)X_n^T = \max_{D_n} \left(D_n X_n^T \right). \quad (1)$$

By the definition of the approximation parameters, for a scheduling algorithm $ALGO$ that approximates a MWM algorithm with approximation parameters $\left(\frac{a}{b}, d\right)$, $a, b \in \mathbb{N}$, a and b prime to each other, there holds

$$D_n(ALGO)X_n^T \geq \frac{a}{b} D_n(MWM)X_n^T - d. \quad (2)$$

We formalize the notion of the stability of a switch [1].

Definition 1 *Let $Y_n = (y_n(1), ..., y_n(M))$ be the row vector of a system of M queues at time n, where $y_n(i)$ is the length of the queue i at time n. A system of queues is said to be strongly stable if*

$$\lim_{n\to\infty} \sup E||Y_n|| < \infty. \quad (3)$$

We state the following theorem:

Theorem 1 *Be given a fixed positive integer m. Given a system of queues whose evolution is described by a discrete time Markov chain with state vector $X_n \in \mathbb{N}^H$, if a lower bounded, steady function $V(Y_n)$, called Lyapunov function, $V : \mathbb{N}^H \to \mathbb{R}$ can be found such that for all $n = ma$, $a \in \mathbb{N}$*

$$E[V(Y_{n+m})|Y_n] < \infty \; \forall \, Y_n,$$

and there exist $\epsilon \in \mathbb{R}^+$ and $B \in \mathbb{R}^+$ such that for all $n = ma$

$$E[V(Y_{n+m}) - V(Y_n)|Y_n] < -\epsilon, \;\; \forall ||Y_n|| > B,$$

then the system of queues is strongly stable.

Proof: In [1], the theorem is stated for the case $m = 1$. For $m \geq 1$, the proof is similar to the proof in [1].

We will use the Lyapunov function $V(X_n) = X_n X_n^T$. Whenever we speak of a matching algorithm of linear complexity, we mean that it is of linear complexity in the number of edges E, where always $|E| = N^2$ in the case of a bipartite $N \times N$ matching as considered in this paper.

3 Approximations to the MWM - algorithm

3.1 The deployment of approximation algorithms in a switch fabric

We consider approximation algorithms that approximate the MWM algorithm with approximation parameters $\left(\frac{a}{b}, d\right)$. To compensate for the factor $\frac{a}{b}$, we propose to deploy all approximation algorithms with a speed-up of $S = \frac{b_1}{a_1} \geq \frac{b}{a}$. We propose two different modes to implement an approximation algorithm in a CIOQ switch.

In the $mode\_keep$, the scheduling algorithm computes a matching at the beginning of a time slot n. It then keeps the matching constant until the beginning of the time slot $n + a_1$, when a new matching is calculated. In the interval $[n, n + a_1)$, up to cells b_1 are forwarded at equally spaced spaced time intervals of length $\frac{a_1}{b_1}$. For a given algorithm $ALGO$, we denote the $mode\_keep$ of the algorithm as $ALGO^k$. For the $mode\_keep$, the evolution of the queue lengths is described as follows:

$$\begin{aligned} X_{n+\frac{da_1}{b_1}} &= X_n - dD_n(ALGO) \\ &\quad + \sum_{0\leq c<\frac{da_1}{b_1}} A_{n+c} + D_\delta, \; 1 \leq d \leq b_1, \quad (4) \end{aligned}$$

where $D_\delta = max\left(\underline{0}, dD_n - X_n - \sum_{0\leq c<\frac{da_1}{b_1}} A_{n+c}\right)$. Here, $\underline{0}$ is the NxN matrix with all entries equal to zero and and the maximum is taken for each matrix entry individually. If $D^{i,j} = 1$, the entry $D_\delta^{i,j}$ denotes the difference between the number of cells that have been forwarded in the interval $[n, n + \frac{da_1}{b_1})$, and the number of internal time slots d in the interval.

In the $mode\_reconfig$, a new matching is computed every internal timeslots, i.e., every $\frac{a_1}{b_1}$ external time slots and cells are forwarded accorded to a calculated matching only once. For a given algorithm $ALGO$, we denote the $mode\_reconfig$ of the algorithm as $ALGO^r$. The queue evolution for the $mode\_reconfig$ is described as follows:

$$\begin{aligned} X_{n+\frac{da_1}{b_1}} &= X_n - \sum_{e=0}^{d-1} D_{n+\frac{ea_1}{b_1}}(ALGO) \\ &\quad + \sum_{0\leq c<\frac{da_1}{b_1}} A_{n+c} + E_\delta, \; 1 \leq d \leq b_1, \quad (5) \end{aligned}$$

where E_δ is an an NxN matrix where each entry is an integer between 0 and b. If at time n, there holds $X_n^{i,j} \geq b_1$, then there is $E_\delta^{i,j} = 0$. If at time n, there holds $X_n^{i,j} < b_1$, then depending on the VOQ length $X_n^{i,j}$ at time n and the arrival patterns $A_{n+c}^{i,j}$, $0 \leq c \leq d-1$, in the interval $[n, n+\frac{da_1}{b_1})$, the switch might not always be able to forward a packet from $VOQ^{i,j}$ even if the scheduling algorithm prescribes so because $D^{i,j}_{n+\frac{da_1}{b_1}} = 1$. The value $E_\delta^{i,j}$ equals the number of instances where this happens for the $VOQ^{i,j}$ and thus takes values c in the range $0 \leq c \leq b_1$.

The $mode\_keep$ mode requires less computations than the $mode\_reconfig$ mode. However, the $mode\_reconfig$ reacts faster to the changing lengths of the VOQ. Following an analysis in [11], one can show that the $mode\_keep$ mode leads to larger average package delays at the $VOQs$ than the $mode\_reconfig$ mode.

3.2 Stability results for mode_keep

We prove the following theorem:

Theorem 2 *Let $\frac{a}{b}$ denote a rational number ≤ 1 and d an absolute constant. We assume a scheduling algorithm $ALGO$ that approximates the MWM algorithm with approximation parameters $\left(\frac{a}{b}, d\right)$. Then the scheduling algorithm $ALGO^k$, when it is deployed with a speed-up $S = \frac{b_1}{a_1} \geq \frac{b}{a}$, stabilizes an CIOQ*

Proof: We set $B_{n,a_1} = \sum_{c=0}^{a_1-1} A_{n+c}$. Using (4), we consider the drift of the Lyapunov function $V(X_n)$ over a time inter-

val $[n, n+a_1)$,

$$\begin{aligned} & E\left[X_{n+a_1}X_{n+a_1}^T - X_nX_n^T|X_n\right] \\ = \; & 2E\left[(B_{n,a_1} - b_1D_n(ALGO) + D_\delta)X_n^T|X_n\right] \\ & +2E\big[(B_{n,a_1} - b_1D_n(ALGO) + D_\delta) \\ & \times(B_{n,a_1} - b_1D_n(ALGO) + D_\delta)^T|X_n\big] \\ \leq \; & 2a_1\lambda X_n^T - 2b_1D_n(ALGO)X_n^T + 2b_1^2N^2 \\ \leq \; & 2a_1\lambda X_n^T - \frac{2b_1a}{b}D_n(MWM)X_n^T + 2b_1^2N^2 + 2b_1d, \end{aligned}$$

where in the last inequality we have used (2). As in [10], we obtain for sufficiently large $||X_n||$

$$E\left[X_{n+a_1}X_{n+a_1}^T - X_nX_n^T|X_n\right] \quad \leq \quad -\epsilon a_1,$$

which implies Theorem 2 using Theorem 1.□

Theorem 2 is a slight improvement over lemma 7 in [1] which requires a speed-up $S \geq \frac{a}{b} + \epsilon$ instead of $\frac{a}{b}$ in Theorem 2. This improvement will allow us to establish stronger stability results for the algorithms discussed in the sequel.

3.3 Stability results for $mode\_reconfig$.

We first prove a lower bound for the weight calculated by an algorithm $ALGO^r$.

Theorem 3 *For any algorithm $ALGO$ that approximates the MWM-algorithm with approximation parameters $\left(\frac{a}{b}, d\right)$ and for any two integers a_1 and b_1, that are prime to each other and that satisfy $\frac{b_1}{a_1} \geq \frac{b}{a}$, there is*

$$\begin{aligned} & \sum_{d=0}^{b_1-1} D_{n+\frac{da_1}{b_1}}(ALGO^r)X_n^T \\ \geq \; & \sum_{c=0}^{a_1-1} D_{n+c}(MWM)X_{n+c}^T - C_{N,b_1,d}, \end{aligned} \qquad (6)$$

where $C_{N,b_1,d}$ is an absolute constant that depends at most on N, b_1, and d. On both sides of the equation (6), the occupancy matrices X_n are assumed to be the occupancy matrices determined by the arrival process and the algorithm $ALGO^r$ in (5).

Proof: We see from (2) and (1) that for any $0 \leq d \leq b_1 - 1$:

$$\begin{aligned} & D_{n+\frac{da_1}{b_1}}(ALGO)X_{n+\frac{da_1}{b_1}}^T \\ \geq \; & \frac{a_1}{b_1}D_{n+\frac{da_1}{b_1}}(MWM)X_{n+\frac{da_1}{b_1}}^T - d \\ \geq \; & \frac{1}{b_1}\sum_{c=0}^{a_1-1} D_{n+c}(MWM)X_{n+\frac{da_1}{b_1}}^T - d. \end{aligned} \qquad (7)$$

We note that because the algorithm $ALGO$ is deployed with a speed-up of $\frac{a_1}{b_1}$, then for any values $n + \frac{da_1}{b_1}$, $0 \leq d \leq b_1 - 1$, and any $n + c$, $0 \leq c \leq a_1 - 1$, the absolute value of the difference of the corresponding entries in the occupancy matrices $X_{n+\frac{da_1}{b_1}}$ and X_{n+c} is at most b_1. This is due to the fact that at most b_1 cells can leave a VOQ in the time interval $[n, n + a_1)$, and at most a_1 cells can arrive at a VOQ in the same time interval. Thus,

$$D_{n+c}(MWM)X_{n+\frac{da_1}{b_1}}^T \quad \geq \quad D_{n+c}(MWM)X_{n+c}^T - b_1N^2,$$

$\forall c$, $0 \leq c \leq a_1 - 1$. Thus, we derive from (7):

$$\begin{aligned} & \sum_{d=0}^{b_1-1} D_{n+\frac{da_1n}{b_1}}(ALGO)X_{n+\frac{da_1n}{b_1}}^T \\ \geq \; & \sum_{c=0}^{a_1-1} D_{n+c}(MWM)X_{n+c}^T - b_1d - b_1N^2. \end{aligned} \qquad (8)$$

(6) follows from (8) because $||X_{n+\frac{da_1n}{b_1}} - X_n|| \leq N^2b_1$. □

We now prove a stability theorem for matching algorithms deployed in $mode\_reconfig$:

Theorem 4 *We assume a matching algorithm $ALGO$ that approximates the MWM algorithm with approximation parameters $\left(\frac{a}{b}, d\right)$. Then the scheduling algorithm $ALGO^r$ when it is deployed with a speed-up $S = \frac{b_1}{a_1} \geq \frac{b}{a}$ stabilizes a CIOQ switch.*

Proof: Using (5), we consider the drift of the Lyapunov function $V(X_n)$ over a_1 time slots:

$$\begin{aligned} & X_{n+a_1}X_{n+a_1}^T - X_nX_n^T \\ = \; & 2\left(\sum_{c=0}^{a_1-1} A_{n+c} - \sum_{d=0}^{b_1-1} D_{n+\frac{da_1}{b_1}}(ALGO) + E_\delta\right)X_n^T \\ + \; & \left(\sum_{c=0}^{a_1-1} A_{n+c} - \sum_{d=0}^{b_1-1} D_{n+\frac{da_1}{b_1}}(ALGO) + E_\delta\right) \\ \times \; & \left(\sum_{c=0}^{a_1-1} A_{n+c} - \sum_{d=0}^{b_1-1} D_{n+\frac{da_1}{b_1}}(ALGO) + E_\delta\right)^T \\ \leq \; & 2\left(\sum_{c=0}^{a_1-1} A_{n+c} - \sum_{d=0}^{b_1-1} D_{n+\frac{da_1}{b_1}}(ALGO)\right)X_n^T + 2b_1^2N^2. \end{aligned}$$

Using (6), we get:

$$\begin{aligned} & X_{n+a_1}X_{n+a_1}^T - X_nX_n^T \\ \leq \; & 2\left(\sum_{c=0}^{a_1-1} A_{n+c}X_n^T - \sum_{d=0}^{a_1-1} D_{n+c}(MWM)X_{n+c}^T\right) \\ & +2b_1^2N^2 + 2C_{N,b_1,d} \\ \leq \; & 2\left(\sum_{c=0}^{a_1-1} A_{n+c} - \sum_{d=0}^{a_1-1} D_{n+c}(MWM)\right)X_{n+c}^T \\ & +3b_1^2N^2 + 2C_{N,b_1,d}, \end{aligned} \qquad (9)$$

where we have used the fact that the corresponding entries of the occupancy matrices X_n and X_{n+c} differ by at most b_1. Taking the expectation with regard to X_n and arguing as in [10], we derive from (9)

$$E\left[X_{n+a_1}X_{n+a_1}^T - X_nX_n^T|X_n\right] \leq -\epsilon \sum_{c=0}^{a_1-1} ||X_{n+c}|| \quad (10)$$

for $||X_{n+c}||$ sufficiently large for all c, $0 \leq c \leq a_1 - 1$.. Noting again that $||X_{n+c} - X_{n+c_1}|| \leq b_1N^2$, we see $E\left[X_{n+a_1}X_{n+a_1}^T - X_nX_n^T|X_n\right] \leq -\epsilon$, which proves the theorem.□

4 Approximation algorithms with a ratio $\frac{1}{2}$.

The most common approximation algorithms to a MWM algorithm are variations of maximal matching algorithms. A maximal matching is a matching that is not properly contained in any other matching of the graph.

The greedy maximal matching algorithm [1], works similar to a general maximal matching as explained in [8], but chooses in each step not an arbitrary but the heaviest edge currently available. It has a running time of $O(E \log E)$ and a performance ratio of $\frac{1}{2}$.

Preis [12] presented another linear time approximation algorithm for a MWM algorithm with a performance ratio of $\frac{1}{2}$. The main idea is to replace the heaviest edge needed by the greedy algorithm with a locally heaviest edge.

A different approach is used by Drake and Hougardy in [5]. The main idea of the proposed algorithm is to grow in a greedy way two matchings independently and to return the heavier of both as a result. Again, this linear time algorithm has a performance ratio of $\frac{1}{2}$. In [6], the same authors use the idea of local improvements to a given matching as a postprocessing step to enhance the performance of the approximation algorithm for the MWM problem in practice. The postprocessing requires linear time, but does not improve the performance ratio of $\frac{1}{2}$.

As all algorithms discussed in this section approximate the MWM algorithm with approximation parameters $\left(\frac{1}{2}, d\right)$, we deduce from Theorems 2 and 4:

Theorem 5 *The approximation algorithms to a MWM algorithm presented in this section stabilize a CIOQ switch when they are deployed in either $mode\_keep$ or $mode\_reconfig$ with a rational speed-up $S \geq 2$.*

Thus, among others, we provide a new way to prove that greedy maximal matching weight matching algorithms are stable with a rational speed-up of $S \geq 2$ as shown in [1].

5 The $improve\_matching$ algorithm

In this section, we first describe the $improve\_matching$ algorithm introduced in [7]. Then we apply our previous results to prove stability theorems for the $improve\_matching$ algorithm.

5.1 Description of the $improve\_matching$ algorithm

We give an overview of the main structure of the algorithm and refer the reader for the missing details to [7].

It is known from graph theory that for any given weight M, the weight of which is not the largest possible, one can replace some edges of M with some edges of the graph such that the new set obtained is again a matching and has a strictly larger value than M. Such a process, which adds a new set $S \subset E$ to M and removes a set $M(S)$ is called an augmentation. The set S is called the augmenting set for this augmentation and the set $M(S)$ is defined as the set of all edges in M that are incident with an end vertex of an edge in S. If S contains edges of M, then these are also contained in $M(S)$. The gain of an augmentation is defined as $w(S) - w(M(S))$ which is the increase of weight achieved by the augmentation.

The idea of the $improve\_matching$ algorithm is first to use standard techniques to expand a given matching to a maximal matching (if the given matching is not already maximal) and then to make local improvements via appropriate augmentations to the maximal matching.

The $improve\_matching$ algorithm considers only local improvements that are obtained via *short augmentations.* A *short augmentation* is defined as an augmentation such that all the edges in the augmenting set are adjacent to some edge $e \in E$. This edge e is called a center of this *short augmentation*. It may or may not belong to the actual matching which is locally changed. Examples of short augmentations are given in [7].

Further, the algorithm does not consider all *short augmentations*, but only those *short augmentations* that lead to a local gain of a factor of at least β, where β is a fixed constant > 1. We call such a short augmentation that satisfies the inequality $w(S) \geq \beta w(M(S))$ a β-augmentation and the augmenting set S the β-augmenting set. In a particular instant, there might be more than one possible β-augmentation. Intuitively, it is desirable to choose the β-augmentation that produces the biggest gain. However, for the purpose of the $improve\_matching$ algorithm, it is sufficient to choose a *good β-augmentation.* A β-augmentation with center e is called *good* if it achieves at least $(\beta - 1)/(\beta - \frac{1}{2})$ fraction of the gain that the best β-approximation with center e can achieve.

We now formally define the $improve\_matching$ algorithm in figure 2. First, the input matching M is made maximal (if necessary) and then no further changes are made to M. Instead, M is copied to M' and all local augmentations are done with respect to M'. The algorithm visits each edge $e \in M$ only once, and if it finds any β-augmentation set at this edge in M', it performs a good β-augmentation centered at e in M'. In [7], it is shown that the the complexity

Algorithm $improve\_matching$
$(G = (V, E), w : E \to \mathbb{R}^+, M)$

```
1    make M maximal
2    M' := M
3    for e ∈ M do begin
4       if there exists a β-augmentation in M' with
           with center e
5       then augment M' by a good β-augmentation
           with center e
6    end
7    return M'
```

$Fig.2$ $The\, improve\_matching\, algorithm$

of the $improve\_matching$ is linear in the number of edges E.

5.2 Application of the $improve\_matching$ algorithm to achieve a performance ratio of $\frac{2}{3} - \epsilon$.

In order to achieve a performance ratio of $\frac{2}{3} - \epsilon$, the $improve\_matching$ algorithm is applied iteratively. We first use a linear time maximal matching algorithm (see section 4) to calculate a maximal matching M_0 (or extend a given matching to a maximal matching) with weight at least as large as $w(M_0) \geq \frac{1}{2} w(M_{opt})$, where M_{opt} denotes a maximum weight matching of the considered graph $G = (V, E)$. We then apply the $improve\_matching$ algorithm to the matching M_0 to obtain a matching M_1 and then iteratively apply the algorithm to the matching M_i to obtain a matching M_{i+1}. It is shown in [7] that for a careful choice of $\beta_i = \beta$, where β_i is chosen differently for each i, there holds $w(M_{i+1}) \geq w_{i+1} w(M_{opt})$, where $w_{i+1} \geq \frac{2}{3} - \frac{16}{3(i+1)}$. Thus, at most $O\left(\frac{1}{\epsilon}\right)$ iterations are required to achieve a performance ratio $\frac{2}{3} - \epsilon$. In summary, we state the following theorem from [7]:

Theorem 6 *The iterated application of the $improve\_matching$ algorithm finds a matching that has a ratio of at least $\frac{2}{3} - \epsilon$ of the maximum weight matching.*

We deduce from Theorems 2, 4, and 6:

Theorem 7 *The iterated $improve\_matching$ algorithm when it is deployed in either $mode\_keep$ or $mode\_reconfig$ with a rational speed-up $S \geq \frac{3}{2} + \epsilon$ stabilizes a CIOQ switch under any admissible traffic.*

6 Conclusions

This paper considers the application of matching algorithms that approximate MWM scheduling algorithms as scheduling algorithms for CIOQ switches. We discuss matching algorithms that approximate MWM algorithms with approximation parameters (c, d). We present two modes $improve\_keep$ and $improve\_reconfig$ for the deployment of the matching algorithms. We show that in both modes, a matching algorithm that approximates a MWM algorithm with approximation parameters $\left(\frac{a}{b}, d\right)$ stabilizes a CIOQ switch when it is deployed with a rational speed-up $S \geq \frac{b}{a}$. These results allow us to prove the stability for four different matching algorithms when they are deployed with a speed-up $S \geq 2$. Finally, we show for the first time the existence of a matching algorithm - the $improved\_matching$ algorithm - that guarantees the stability of a switch when it is deployed with a rational speed-up $S \geq \frac{3}{2} + \epsilon$.

References

[1] Ajmone Marsan, M., Leonardi, E., Mellia, M., Neri, F., *On the stability of input-buffer cell switches with speed-up,* Proc. Infocom 2000, Tel Aviv.

[2] Bauer, C., *Distributed scheduling policies in networks of input-queued packet switches,* ACM Comp. Commun. Review, Vol. 34, no. 3, July 2004, 83 -92.

[3] Benson, K., *Throughput of crossbar switches using maximal matching algorithms,* Proc. of IEEE ICC 2002, New York City.

[4] Dai, J.G., Prabhakar, B., *The throughput of data switches with and without speedup,* Proc. of IEEE Infocom 2000, Tel Aviv.

[5] Drake, D.E., Hougardy, S., *A simple approximation algorithm for the weighted matching problem,* Information Processing letters 85 (2003), 211-213.

[6] Drake, D.E., Hougardy, S., *Linear time local improvements for weighted matchings in graphs,* WEA 2003, LNCS 2647, Seiten 107-119, 2003.

[7] Drake, D. E., Hougardy, S., *A linear time approximation algorithm for weighted matchings in graphs,* preprint: http://www.informatik.huberlin.de/ hougardy/paper.html.

[8] Gabow, H.N., Tarjan, R.E., *Faster scaling algorithms for general graph-matching problems,* J. ACM 38:4, 1991, 815-853.

[9] Gabow, H.N., *Data structures for weighted matching and nearest common ancestors with linking,* SODA 1990, 434 - 443.

[10] Keslassy, I., McKeown, N., *Achieving 100% throughput in an input queued switch,* IEEE Trans. on Communications, vol. 47, no. 8, Aug. 1999, 1260 - 1272.

[11] M. J. Neely, E. Modiano, and C.E. Rohrs, *Tradeoffs in Delay Guarantees and Computation Complexity for NxN Packet Switches,* Proc. of the Conf. on Information Sciences and Systems, Princeton: March 2002.

[12] Preis, R., *Linear time 1/2 approximation algorithm for maximum weighted matching in general graphs,* Symposium on Theoretical Aspects of Computer Science (STACS) 1999, Springer LNCS 1563, 259 - 269.