Coarse Grained Parallel Algorithm for Hamiltonian Circuit in Convex Bipartite Graphs

Marco A. Stefanes\textsuperscript{1}, Diego P. Rubert\textsuperscript{1}, and José Soares\textsuperscript{2}

\textsuperscript{1}Faculdade de Computação, Federal University of Mato Grosso do Sul, Campo Grande-MS, Brazil
\textsuperscript{2}Department of Computer Science, University of S. Paulo, S. Paulo-SP, Brazil

Abstract—A bipartite graph $G = (V, W, E)$ is convex if there exists an ordering of the vertices of $W$ such that, for each $v \in V$, the neighbors of $v$ are consecutive in $W$. In this work, we address the Hamiltonian Circuit Problem, a well-known problem in Combinatorial Optimization. We present a novel sequential linear-time algorithm for determining a Hamiltonian circuit in convex bipartite graphs which can be easily parallelized. We also describe a coarse grained parallel algorithm for that problem which runs in time $O((|V|/p) \log(|V|/p) \log p)$, for $p$ processors, using $O(\log p)$ communication rounds. We also show how to efficiently implement our solution into PRAM and coarse grained parallel models. Our algorithm provides parallel scalability on commodity clusters. We have made experiments in a cluster composed of 64 processors, obtaining increasing speedups in our implementation. As far as we know, that is the first coarse grained parallel algorithm for the problem.

Keywords: Scalable parallel algorithms, Convex bipartite graphs, Linear-time Hamiltonian circuit, Coarse grained parallel computing

1. Introduction

PRAM – parallel random access machine – is the main theoretical model of parallel computing. However, for many problems PRAM algorithms do not attain expected results in practice. One of the reasons is that fine grained PRAM model does not consider accordingly communication and computation costs. Features of commercially available parallel computers have shown we should take into account that communication costs are different of processing costs during the design of algorithms. Recently, several problems are implemented in more practical models which have been tested in cluster systems \cite{1}, \cite{2}, \cite{3}, \cite{4}, in cloud environments \cite{5} and Hybrid environments \cite{6}, where it is possible to explore parallelism in multiple levels. An important feature of such parallel algorithms is its scalability, that is, the ability to handle efficiently a growing amount of work. We refer to such models as coarse grained parallel models. They are very appropriate to current parallel machines, since the computation speed is faster than communication speed in such machines.

Bipartite convex graphs were introduced by Glover \cite{7}, initially motivated by some industrial applications. Nowadays algorithms have been developed for this class of graphs in many applications, such as scheduling problems \cite{8}, DNA analysis \cite{9} and constraint programming \cite{10}. Several other applications of convex bipartite graphs were described in \cite{11}, \cite{12}, \cite{13}. Sequential and parallel algorithms (\cite{11}, \cite{8}, \cite{14}, \cite{15}) have been developed for finding a maximum matching in those graphs. A maximum matching is also used in \cite{11}, \cite{16}, \cite{17} for determining a maximum independent set in convex bipartite graphs. In this paper, we use a maximum matching algorithm for finding a Hamiltonian circuit.

A Hamiltonian circuit in a graph $G = (V, E)$ is a simple circuit $(v_1, \ldots, v_n, v_1)$, $n = |V|$, containing each vertex of $V$ exactly once. If a graph contains a Hamiltonian circuit, then it is called a Hamiltonian graph. The problem of deciding whether a graph is Hamiltonian or not is a well-known topic in graph theory. The problem is NP-complete \cite{18} even for bipartite graphs \cite{19} or chordal bipartite graphs \cite{20}.

Let $G = (V, W, E)$ be a bipartite graph, where $(V, W)$ is the partition of the vertices and $E$ the set of edges. We say that $G$ is convex if there is an ordering “$\leq$” of the vertices of $W$ such that, for each vertex $v \in V$, the neighbors of $v$ are consecutive in $W$. Considering the ordering of $W$, $G$ can be represented by a set of $|V|$ pairs $\{(\text{begin}(v), \text{end}(v)) | v \in V\}$, where $\text{begin}(v) \in W$ and $\text{end}(v) \in W$ is the smallest (largest) vertex of $W$ adjacent to $v$. This is called a compact representation of $G$. Throughout this work convex bipartite graphs are given by their compact representation.

A goal of this work is to determine a Hamiltonian circuit in convex bipartite graphs. We present an $O(|V|)$ sequential algorithm for the problem and its version in PRAM and coarse grained parallel models. As far as we know, that is the first coarse grained parallel algorithm for the problem. Müller describes in \cite{20} an $O(|V|^2)$ algorithm for the same problem. In a more general class of graphs, the circular convex bipartite graphs, Liang and Blum \cite{12} present a sequential algorithm $O(|V|)$ for finding Hamiltonian circuits. However, our sequential algorithm has the feature of being easily parallelizable, as showed in this work. By experiments made in a cluster composed of 64 processors, we show the algorithm is scalable, since its performance improves proportionally to the growing in the number of processors. We obtained increasing speedups ranging from 0.71 to 3.92 in such machines.

The remainder of this work is structured as follows. First
of all, we specify the models of parallel computing in Section 2. In Section 3 we describe an $O(|V|)$ sequential algorithm for deciding whether a convex bipartite graph is Hamiltonian. The complexity and correctness of the algorithm is presented in section 4. In Section 5 we detail how to implement the algorithm in the PRAM and BSP/CGM parallel models. Then, in Section 6 we report some experimental results. Finally, in Section 7, we present some conclusions.

2. Parallel Models

PRAM [21] is the best known parallel model, which is a straightforward generalization of the Von Neumann model. It is an idealized theoretical model of a shared memory MIMD machine. Although PRAM model is important from theoretical viewpoint, speedups obtained in practical implementations do not correspond to that expected in available parallel computers. There is a more practical model: BSP/CGM which is a coarse grained parallel model. In this model the communication between processors is done via an arbitrary network. Under the BSP [22] (Bulk Synchronous Parallel), computations are organized as a sequence of supersteps separated by synchronization barriers. The parameters of this model are: $p$, the number of processors; $L$, the minimum time of a superstep; and $g$, the quotient between speed of local computation and network bandwidth. The model CGM [23] (Coarse Grained Multicomputer) is a version of the BSP consisting of $p$ processors with $O(n/p)$ local memory each, where $n/p \geq p'$, for some $\epsilon > 0$. A BSP/CGM algorithm consists of local computation alternated with global communication. In a communication round, each processor sends and receives $O(n/p)$ data items.

3. Sequential Algorithm

The Algorithm Hamiltonian Circuit and its parallel versions are closely related to greedy maximum matchings. A matching $M$ in a graph $G$ is a subset of the edges of $G$ such that no two edges in $M$ are incident to a same vertex of $G$. A matching is maximum if its cardinality (number of edges) is as large as possible. A vertex $v$ is matched by $M$ if there exists an edge in $M$ incident to $v$. A vertex $v$ is free with respect to $M$ if $v$ is not matched by $M$. If there is no free vertex with respect to a matching $M$, it is called a perfect matching. The following concept, which uses the ordering “$\leq$” of $W$, has been shown to be very useful for determining a maximum matching in bipartite convex graphs.

Definition 3.1: A matching $M$ in a bipartite convex graph $G = (V, W, E)$ is greedy if it has the following two properties:
1) if $(v, w) \in M$ and $v \in V$, then, for each $w' \in W$, with $\text{begin}(v) \leq w' < w$, there exists $v' \in V$ such that $(v', w') \in M$ and $\text{end}(v') \leq \text{end}(v)$;
2) if $w \in W$ is connected to a free vertex $v' \in V$, then there exists $v \in V$ such that $(v, w) \in M$ and $\text{end}(v) \leq \text{end}(v')$.

An important property of greedy matchings was discovered by Glover [7]: every greedy matching is also a maximum matching.

In Hamiltonian convex bipartite graphs we will show how to find a Hamiltonian circuit in convex bipartite graphs consisting, essentially, of the disjoint union of two perfect matchings. Although every Hamiltonian circuit in a bipartite graph is a union of two disjoint perfect matchings, it is not true that the union of two disjoint perfect matchings is always a Hamiltonian circuit. This fact is illustrated by the graph $G_A$ in the Figure 1. The union of the two disjoint perfect matchings $\{(v_1, w_1), (v_2, w_2), (v_3, w_3), (v_4, w_4)\}$ and $\{(v_1, w_2), (v_2, w_1), (v_3, w_4), (v_4, w_3)\}$ is the union of two disjoint circuits. Depending on the choice of the first perfect matching, the remaining graph may not even contain another perfect matching. This fact is illustrated by the graph $G_B$ in the Figure 1. The matching $\{(v_1, w_1), (v_2, w_2), (v_3, w_3)\}$ leaves the graph without another perfect matching.

As before we assume that a graph $G = (V, W, E)$ is given by its compact representation. We assume that the vertices of $W$ are labeled is such a way that the sequence $w_1, w_2, \ldots, w_m$ are the vertices of $W$ listed according to the ordering “$\leq$”. We also define, for each $w \in W$, $\text{next}(w)$ as the vertex after $w$ according to that order. Note that $\text{next}(w_m)$ is undefined. Let $M$ be a matching in $G$. If $v \in V$ and $(v, w) \in M$, we denote by $M(v)$ the vertex $w$. For $X \subseteq V$, we denote by $N(X) \subseteq W$ the set of vertices in $W$ adjacent to vertices in $X$. Again, let $n = |V|$.

Algorithm Hamiltonian Circuit

Input: A graph $G = (V, W, E)$ in its compact representation.

Output: A Hamiltonian circuit if $G$ is Hamiltonian.

(1) If $|V| \neq |W|$, then return “$G$ is not Hamiltonian.”
(2) Let $v_1$ be any vertex in $V$ adjacent to $w_1$. $G_1 := G - v_1 \cdot w_1$.
(3) Find a greedy matching $M$ in $G_1$.
(4) If $|M| > n - 1$, then return “$G$ is not Hamiltonian.”
(5) $G_2 := G - w_1$.
(6) For $v_i \in V \setminus \{v_1\}$ do $\text{begin}(v_i) := \text{next}(M(v_i))$. 

Fig. 1: Two Hamiltonian convex bipartite graphs illustrating the relationship between Hamiltonian circuit and perfect matching.
If, for some \( v_i \), \( \text{next}(M(v_i)) \) is undefined, then return “\( G \) is not Hamiltonian.”

(7) Find a greedy matching \( M' \) in \( G_2 \).

(8) Let \( v_k \in V \) be a free vertex with respect to \( M' \). If \( |M'| < n - 1 \) or \( w_n \notin N(v_k) \), then return “\( G \) is not Hamiltonian.” Else, return \( M \cup M' \cup (v_1, w_1) \cup (v_k, w_n) \).

Figures 2, 3, 4 and 5 describe an example of a running of the algorithm beginning by the choice of the vertex \( v_1 \) in the Step 2, the greedy matching \( M \) found in \( G_1 \) in the Step 3, the greedy matching \( M' \) found in \( G_2 \) in the Step 7 and the resulting Hamiltonian circuit.

The Hamiltonian circuit found by the algorithm consists of either the circuit

\[
(\ v_1, v_1, M'(v_1), M(M'(v_1)), M'(M(M'(v_1))), \ldots, v_k, w_n, \ M'(w_n), M(M'(w_n)), M'(M(M'(w_n))), \ldots, M(w_1), w_1 )
\]

or the circuit

\[
(\ v_1, v_1, M'(v_1), M(M'(v_1)), M'(M(M'(v_1))), \ldots, w_n, v_k, \)
\]

4. Correctness and Complexity

In this section, we demonstrate the correctness and time complexity of the Algorithm Hamiltonian Circuit. The following lemmas are useful to show the correctness of the algorithm.

Lemma 4.1: If \( G = (V, W, E) \) is a Hamiltonian bipartite graph, then, for any proper non-empty subset \( S \) of \( V \), we have \( |S| < |N(S)| \).

Proof. Let \( C \) be a Hamiltonian circuit in \( G \). Since \( S \) is a non-empty proper subset of \( V \), considering only the edges in \( C \) incident to vertices in \( S \), we find a neighborhood of \( S \) whose cardinality is larger than \( |S| \).

Lemma 4.2: If \( G \) is Hamiltonian, then \( G_1 \) contains a perfect matching.

Proof. Let \( C \) be a Hamiltonian circuit in \( G \). Let \( P \) be the path from \( v_1 \) to \( w_n \) in \( C \), and \( P' \) be the path from \( w_n \) to \( v_1 \) in \( C \). Notice that, since the graph is bipartite, both \( P \) and \( P' \) have odd number of edges. The paths \( P - v_1 - w_n \) and \( P' - v_1 - w_n \) are vertex disjoint and their edges contain a perfect matching in \( G_1 = G - v_1 - w_n \).

For the sake of analysis, we consider next that vertices in \( V \) are labeled in such a way that for each \( i \), \( 1 < i \leq n \), \( M(v_i) = w_{i-1} \).

Lemma 4.3: If \( G \) is Hamiltonian, then \( M(v_j) < \text{end}(v_j) \) for each \( j \), \( 1 < j \leq n \).

Proof. By contradiction, suppose that \( G \) is Hamiltonian and there exists \( 1 < j \leq n \) such that \( M(v_j) = w_{j-1} \geq \text{end}(v_j) \). Since \( v_j \) is matched to \( w_{j-1} \), there can only be the case that \( M(v_j) = \text{end}(v_j) \). If \( N(\{v_1, \ldots, v_{j-1}\}) \subseteq \{w_1, \ldots, w_{j-1}\} \), we have, by Lemma 4.1, \( G \) is not Hamiltonian, a contradiction. So, we may assume that there exists \( r < j \) such that \( \text{end}(v_r) > \text{end}(v_j) \). We choose \( r \) as large as possible, in such a way that, for each \( i \), \( r < i \leq j \), we have that \( \text{end}(v_i) \leq \text{end}(v_j) \). From the above observation, and remembering that \( M \) is greedy, it holds that
By construction of greedy, we have that $N(\{v_{r+1}, \ldots, v_j\}) = \{M(v_{r+1}), \ldots, M(v_j)\}$, and, by Lemma 4.1, $G$ is not Hamiltonian. A contradiction.

**Lemma 4.4:** If $G$ is Hamiltonian, then each vertex in $W - \{w\}$ is matched by $M'$.  

**Proof.** Since $M = \{(v_2, w_1), (v_3, w_2), \ldots, (v_n, w_{n-1})\}$ is a matching, using the Lemma 4.3 and the convexity of $G$, $M'' = \{(v_2, w_2), (v_3, w_3), \ldots, (v_n, w_n)\}$ is also a matching. Notice that $M''$ is a matching in $G_2$ that matches each vertex in $W - \{w\}$. Since $M$ is a maximum matching, $M'$ will also match each vertex in $W - \{w\}$. 

If the graph $G$ is Hamiltonian, then, after Step 7 of the algorithm, the graph $G$ contains two matchings $M$ and $M'$, both of them with size $n - 1$. Since in $G_2$ we have $|V| = n$ and $|W| = n - 1$, there exists exactly a free vertex in $V$ with respect to $M'$.

**Lemma 4.5:** Let $v_k$ be the free vertex of $V$ in $M'$. If $G$ is Hamiltonian, then $v_k$ is adjacent to $w_n$.  

**Proof.** By contradiction, suppose that $v_k$ is not adjacent to $w_n$. Let $q$ be such that $end(v_k) = M(v_q) = w_{q-1}$. Since by Lemma 4.3 $end(v_k) > M(v_k)$, we have that $k < q$.

First we show that, for each $i, 1 \leq i < q, M'(v_i) \leq M(v_i)$. By construction of $G_2$, the only vertices in $G_2$ that can be adjacent to vertices in $W^* := \{w_2, w_3, \ldots, w_{q-1}\}$ are the vertices in $V^* := \{v_1, v_2, \ldots, v_{q-1}\}$. As $v_k$ is one of those vertices and, by Lemma 4.4, all the vertices in $W - \{w\}$ are matched by $M'$, $M'$ induces a bijection between $W^*$ and $V^* - w_k$.

By Lemma 4.1, we have that $|N(V^*)| > |\{w_2, w_3, \ldots, w_{q-1}\}|$. Hence, there exists an $r < q$ such that $end(r) > w_{q-1}$. Choose $r$ with that property in such a way that $j$ is maximum, where $w_j := M'(v_j)$. As $end(v_k) < end(v_i)$, $v_k$ is free in $M'$ and $M'$ is greedy, we have that $w_j$ is not adjacent to $v_k$ in $G_2$. Since $w_j = M'(v_j) \leq M(v_j)$ and the neighbors of $v_k$ in $G_2$ are $w_k, w_{k+1}, \ldots, w_{q-1}$, we obtain that $w_j < w_k$.

We now show that $j = r$. Since $w_j = M'(v_j)$, by construction of $G_2$, we have that $j \geq r$. If $j > r$, since $v_j$ is adjacent to $w_j$ in $G_2$ and $M'$ is greedy, we can conclude that $end(v_j) \geq end(v_r)$. This would contradict the choice of $r$, because $M'(v_j)$ would be larger than $M'(v_r)$. So, $j = r$.

Notice that there exist $r - 1$ vertices smaller than $v_r$ matched by $M'$. So, one of them is matched to some vertex not in the set $\{w_2, w_3, \ldots, w_{q-1}\}$. Let $v_k$ be such vertex. It follows that $M'(v_k) > M'(v_r) = w_r$. Since $v_k$ is not matched by $M'$ to $w_r$, we have that $end(v_k) \geq end(v_r)$. Again, this contradicts the choice of $r$, showing that there exists no such $r$. A contradiction, proving the lemma.

Using the lemmas above, we are ready to prove the following theorem. So, we conclude the algorithm correctly compute the Hamiltonian Circuit.

**Theorem 4.6:** The Algorithm Hamiltonian Circuit solves the problem of the Hamiltonian circuit in convex bipartite graphs.  

**Proof.** We will show that the edge set $E_C := M \cup M' \cup (v_1, w_1) \cup (v_k, w_n)$ are exactly the edges of a Hamiltonian circuit in $G$.

Firstly observe that, by construction, $(v_1, w_1)$ is an edge in $G$. The existence of the matchings $M$ and $M'$ and the edge $(v_k, w_n)$ are guaranteed by Lemmas 4.2, 4.4, and 4.5.

By inspection, we can verify that $E_C$ induces a subgraph of $G$ where each vertex has even degree. Hence, $E_C$ induces a collection of disjoint circuits in $G$. We need to show that $E_C$ induces exactly one circuit, and, therefore, a Hamiltonian circuit.

Let $C$ be one of the circuits induced by $E_C$. We shall show that the edge $(v_k, w_n)$ is used by $C$. Since the choice of $C$ is arbitrary and the circuits are disjoint, $E_C$ has to induce exactly 1 circuit.

Since $C$ contains at least 4 edges, the circuit $C$ has to use at least one edge of $M$. So, we may consider that $C = (w_{i_1}, v_1, w_{i_2}, v_1, \ldots, v_1, w_n)$, where $M(w_{i_1}) = v_1$. We will prove that the sequence $j_1, j_2, j_3, \ldots$ is increasing, up to the edge $(v_k, w_n)$ is used.

Recall that the vertices of $W$ were labeled in such a way that for each $i, M(v_i) = w_{i-1}$. So, it holds that $i_1 = j_1 + 1$, showing that $j_2 < i_1$. Next, we show that $i_2 \leq j_2$. If $(v_{i_1}, w_{i_2}) = (v_1, w_n)$, $C$ uses the edge $(v_k, w_n)$, as wished.

So, we may assume that $(v_{i_1}, w_{i_2}) \neq (v_k, w_n)$. Since $i_1 > 1$, it cannot be the case that $(v_{i_1}, w_{i_2}) = (v_1, w_1)$. Hence, it is true that $M'(v_{i_1}) = w_{j_2}$. As $M'$ is a matching in $G_2$, where for each $i, begin_{G_2}(v_i) = next(M(v_i))$, it holds that $M'(v_{i_1}) = w_{j_2} \geq begin_{G_2}(v_{i_1}) = next(M(v_{i_1})) = next(w_{i_1} - 1) = w_{i_1}$. It follows that $i_2 \leq j_2$.

Applying the same idea of the above paragraph, we can show that $j_2 < i_2 \leq j_3$, unless the edge $(v_k, w_n)$ is used.

Since the circuit finishes in $w_{j_2}$, the sequence of indices cannot be always increasing and the edge $(v_k, w_n)$ has to be used by $C$.

The time complexity of the algorithm is dominated by the Steps 3 and 7, which consists in finding greedy matchings in a convex bipartite graph. Lipski and Preparata [11] presented an algorithm for finding a greedy matching in convex bipartite graph which can run in time $O(|V|)$, provided that a special version of the union-find algorithm is used [24]. Summarizing, we have the following theorem.

**Theorem 4.7:** The Algorithm Hamiltonian Circuit solves the problem of the Hamiltonian circuit in convex bipartite graphs in linear-time.
5. Parallel Algorithms

The sequential algorithm from the previous section consists essentially of finding two greedy matchings in a convex bipartite graph \( G = (V, W, E) \). Our parallel version consists in distributing the input graph uniformly among the processors. So, in steps 3 and 7 of the Algorithm Hamiltonian Circuit, we call a parallel greedy matching procedure. The remain of the algorithm stays the same as before. In order to do the steps 3 and 7 in the PRAM model, Dekel e Sahni [8] developed a EREW PRAM algorithm for the maximum matching problem that runs in time \( O(\lg^2 n) \) using \( O(n) \) processors, where \( n = |V| \). In coarse grained parallel computing we can solve the maximum matching problem using the algorithm described by Soares and Stefanes [25] or the algorithm presented by Chan et al. [15] which use \( p \) processors, \( O(\lg p) \) communication rounds, and spend \( O(n \lg p)/p \) time of local computations. The maximum matching found by those algorithms are greedy. Therefore, we have the following theorems.

**Theorem 5.1:** Given a convex bipartite graph \( G = (V, W, E) \), the Hamiltonian circuit problem in \( G \) can be solved in the EREW PRAM model in time \( O(\lg^2 |V|) \), using \( |V| \) processors.

Another solution for the Hamiltonian Circuit described in PRAM model and spending \( O(\lg |V|) \) time, using \( |V|^2 \) processors is given by Bertossi and Bonuccelli [26].

**Theorem 5.2:** Given a convex bipartite graph \( G = (V, W, E) \), the Hamiltonian circuit problem in \( G \) can be solved in the BSP/CGM model using \( p \) processors, \( O(\lg p) \) communication rounds and \( O((|V| \lg p)/p) \) time of local computations. \( \square \)

6. Experimental Results

Our experiments were carried out in a high performance cluster at UFMS (Federal University of Mato Grosso do Sul) using 32 nodes. Each node is a 2.5Ghz quad core Xeon with 2GB of memory. Communication was carried out using a fat tree network implemented over an optical Myrinet switch with 10Gbits/s links.

6.1 Testing Graphs

For each convex bipartite graph \( G = (V, W, E) \) we have used in tests, we generated its compact representation as follow. After choose the size of set \( W \), for each vertex \( v \in V \), we choose randomly the middle of the interval \([\text{begin}(v), \text{end}(v)]\) and with a previously defined density we determine the interval length derived from the Poisson distribution.

In order to test the implementation, we create graphs with intervals using density of 8%, i.e., we generated intervals with average length of 8% of the size of \( W \). That value was chosen aiming to test some special details of the algorithm. We mean, in the Algorithm Hamiltonian Circuit, the interval density of 8% generates, in general, a Non-Hamiltonian graph and, with high probability, with only one matching the algorithm can decide whether the input graph is Hamiltonian or not. In another tests, we generated graphs using interval density of 45%, which enforce the algorithm to invoke matching procedure twice.

6.2 Results

The results related in this section were obtained from average time of ten executions of each instance. The size of the graphs was chosen aiming to equilibrate processing and communication costs, since we observed the algorithm speedup improves as the graph size grows.

![Fig. 6: Parallel algorithm speedups × number of processors for Hamiltonian Circuit, labels represent size of V.](image)

Figure 6 shows the speedups of the algorithm in the cluster for graphs of sizes \( 1 \times 10^7 \), \( 5 \times 10^7 \) and \( 1 \times 10^8 \). Speedups obtained by the algorithm ranged from 0.71 to 3.92 using 64 processors. We can see the algorithm is scalable when number of processors increases, although the speedup was less than 1 for 2 and 4 processors due to the cost of parallelization.

When sequential algorithm has time complexity \( O(n) \) or \( O(n \lg n) \), parallel version of this algorithm tends to have communication cost higher compared to local processing time, mainly when the communication rounds are increasing with the number of processors. The Hamiltonian Circuit Problem has that feature as we can see in Table 1, where we show the processing time compared to communication time.

In the Figures 7 and 8, we see the running time of the Hamiltonian Circuit algorithm for two different kinds of instances, respectively. generated graphs with density of 8% and 45% of the neighborhood for each \( v \in V \).
Table 1: Processing time and communication time, in seconds, for Hamiltonian Circuit Algorithm where $|V| = 1 \times 10^7$, $5 \times 10^7$ and $1 \times 10^8$.

<table>
<thead>
<tr>
<th>Proc</th>
<th>Com</th>
<th>Proc</th>
<th>Com</th>
<th>Proc</th>
<th>Com</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>31.7</td>
<td>170.3</td>
<td>41.9</td>
<td>309.1</td>
<td>80.6</td>
</tr>
<tr>
<td>4</td>
<td>21.0</td>
<td>113.0</td>
<td>54.0</td>
<td>204.1</td>
<td>101.4</td>
</tr>
<tr>
<td>8</td>
<td>12.9</td>
<td>70.0</td>
<td>45.8</td>
<td>126.0</td>
<td>85.9</td>
</tr>
<tr>
<td>16</td>
<td>7.6</td>
<td>41.3</td>
<td>36.7</td>
<td>74.3</td>
<td>69.6</td>
</tr>
<tr>
<td>32</td>
<td>4.4</td>
<td>23.8</td>
<td>30.1</td>
<td>42.8</td>
<td>56.5</td>
</tr>
<tr>
<td>64</td>
<td>2.5</td>
<td>13.7</td>
<td>24.4</td>
<td>24.7</td>
<td>46.0</td>
</tr>
</tbody>
</table>

Graphs with density of 8% of neighborhood

Fig. 7: Total running time in seconds $\times$ number of processors for Hamiltonian Circuit using interval density of 8%, labels represent $|V|$.

As shown in results, the running time of Hamiltonian Circuit Algorithm is basically the maximum matching running time. In the first case, displayed in Figure 7, the input graph is not Hamiltonian and the algorithm performs only one matching in order to make a decision, and in the second case, shown in Figure 8, the input graphs are Hamiltonian and the algorithm calls the maximum matching algorithm twice before making a decision.

7. Concluding Remarks

In this paper, we have investigated parallel solutions to the Hamiltonian Circuit Problem in convex bipartite graphs. We described a linear-time algorithm for finding a Hamiltonian circuit in $G = (V, E)$ that can be implemented under BSP/CGM model using $p$ processors, $O(\lg p)$ communication rounds and $O((|V| \lg p)/p)$ time. We show through practical experiments our algorithm has good scalability. We also presented an algorithm where the Hamiltonian circuit problem can be solved in the EREW PRAM model in time $O(\lg^2 |V|)$, using $|V|$ processors. A frequently considered hierarchy of graph classes is: permutation bipartite graphs $\subset$ doubly convex bipartite graphs $\subset$ convex bipartite graphs $\subset$ chordal bipartite graphs [27]. It would be interesting to develop parallel algorithms to the same problem for the class of chordal bipartite graphs. As future work, we could suggest the analysis of parallel Maximum Edges Bicliques Problem in this class of graphs. Besides, the study of the problem addressed here in more general classes of graphs, for instance, interval graphs, seem us appropriated.

References


