

A parallel algorithm for computing the critical independence number and related sets

Ermelinda DeLaViña *

*Department of Computer and Mathematical Sciences
University of Houston—Downtown, Houston, TX 77002*

Craig E. Larson †

*Department of Mathematics and Applied Mathematics
Virginia Commonwealth University, Richmond, VA 23284*

Received 9 July 2010, accepted 10 May 2012, published online 28 October 2012

Abstract

An independent set I_c is a *critical independent set* if $|I_c| - |N(I_c)| \geq |J| - |N(J)|$, for any independent set J . The *critical independence number* of a graph is the cardinality of a maximum critical independent set. This number is a lower bound for the independence number and can be computed in polynomial-time. The existing algorithm runs in $\mathcal{O}(n^{2.5} \sqrt{m/\log n})$ time for a graph G with $n = |V(G)|$ vertices and m edges. It is demonstrated here that there is a parallel algorithm using n processors that runs in $\mathcal{O}(n^{1.5} \sqrt{m/\log n})$ time. The new algorithm returns the union of all maximum critical independent sets. The graph induced on this set is a König-Egerváry graph whose components are either isolated vertices or which have perfect matchings.

Keywords: Critical independent set, critical independence number, independence number, matching number, König-Egerváry graph.

Math. Subj. Class.: 05C69

1 Introduction

A new faster parallel algorithm is given for finding maximum critical independent sets and calculating the critical independence number of an arbitrary graph.

*Work supported in part by the United States Department of Defense and resources of the Extreme Scale Systems at Oak Ridge National Laboratories.

†Corresponding author.

E-mail addresses: delavinae@uhd.edu (Ermelinda DeLaViña), clarson@vcu.edu (Craig E. Larson)

The following notation is used throughout: the vertex set of a graph G is $V(G)$, the order of G is $n = n(G) = |V(G)|$, the set of neighbors of a vertex v is $N_G(v)$ (or simply $N(v)$ if there is no possibility of ambiguity), the set of neighbors of a set $S \subseteq V(G)$ in G is $N_G(S) = \cup_{u \in S} N(u)$ (or simply $N(S)$ if there is no possibility of ambiguity), the set $N[S] = N(S) \cup S$, and the graph induced on S is $G[S]$. All graphs are assumed to be finite and simple.

A set $I \subseteq V(G)$ of vertices is an *independent set* if no pair of vertices in the set are adjacent. The *independence number* $\alpha = \alpha(G)$ is cardinality of a maximum independent set (MIS) of vertices in G . An independent set of vertices I_c is a *critical independent set* if $|I_c| - |N(I_c)| \geq |J| - |N(J)|$, for any independent set J . A *maximum critical independent set* (MCIS) is a critical independent set of maximum cardinality. The *critical independence number* of a graph G , denoted $\alpha' = \alpha'(G)$, is the cardinality of a maximum critical independent set. If I_c is a maximum critical independent set, and so $\alpha'(G) = |I_c|$, then clearly $\alpha' \leq \alpha$. The *critical difference* d is $\max\{|I_c| - |N(I_c)| : I_c \text{ is an independent set}\}$.

Critical independent sets are of interest for both practical and theoretical reasons. By a theorem of Butenko and Trukhanov, any critical independent set can be extended to a maximum independent set [4]. Zhang and Ageev gave polynomial-time algorithms for finding critical independent sets [17, 1]. Thus, finding a critical independent set is a polynomial-time technique for reducing the search for the well-known widely-studied NP-hard problem of finding a maximum independent set in a graph [7]. Maximum critical independent sets are central in the investigation of the structure of maximum independent sets, a connection via the Independence Decomposition Theorem, recounted in the next section.

The existing algorithm for finding a MCIS and calculating α' runs in $\mathcal{O}(n^{2.5} \sqrt{m/\log n})$ time [11]. It is demonstrated here that there is a parallel algorithm using n processors that runs in $\mathcal{O}(n^{1.5} \sqrt{m/\log n})$ time. The new algorithm finds the set H of vertices which are in *some* maximum critical independent set, that is, the union of all MCISs. The graph induced on this set is a König-Egerváry graph whose non-trivial components each have a perfect matching.

2 The set H of vertices in some MCIS

The correctness of the algorithm presented in the next section depends on the following decomposition theorem, a corollary of, and equivalent to, the Independence Decomposition Theorem in [12]. A *matching* in a graph is a set of pairwise non-incident (or independent) edges. The *matching number* μ of a graph is the cardinality of a maximum matching. A König-Egerváry graph is one where $\alpha + \mu = n$.

Theorem 2.1. (Larson, [12]) *For any graph G , there is a unique set $X \subseteq V(G)$ such that*

1. $\alpha(G) = \alpha(G[X]) + \alpha(G[X^c])$,
2. $G[X]$ is a König-Egerváry graph,
3. for every non-empty independent set I in $G[X^c]$, $|N(I)| > |I|$, and
4. for every maximum critical independent set J_c of G , $X = J_c \cup N(J_c)$.

According to the theorem there is a unique set $X \subseteq V(G)$ such that, for any maximum critical independent set I_c , $I_c \cup N(I_c) = X$. For any graph G let $X = X(G)$ be the set guaranteed by Theorem 2.1. Call $G[X]$ the *distinguished König-Egerváry subgraph* of G . König-Egerváry graphs were first characterized by Deming [6] and Sterboul [16] in the

1970s. The first author's Graffiti.pc program conjectured (number 329 in [5]) a characterization in terms of the critical independence number: a graph G is a König-Egerváry graph if, and only if, $\alpha(G) = \alpha'(G)$. The conjecture was first proven by Larson in [12]. In [14] Levit & Mandrescu extended the statement of this result as follows.

Theorem 2.2. (Levit & Mandrescu, [14]) *The following are equivalent:*

1. G is a König-Egerváry graph,
2. there is a maximum independent set of G that is a MCIS,
3. every maximum independent set of G is a MCIS,

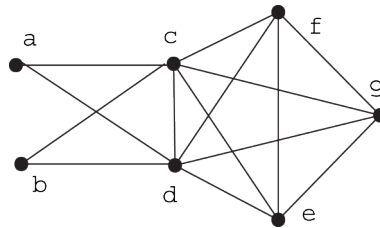


Figure 1: The vertices $I_c = \{a, b\}$ form a (maximum cardinality) critical independent set; this set of vertices can be extended to a maximum independent set of the graph. According to Theorem 2.1 the sets $X = I_c \cup N(I_c) = \{a, b, c, d\}$ and $X^c = V \setminus X = \{e, f, g\}$ induce a decomposition of the graph into a König-Egerváry subgraph $G[X]$ and one, $G[X^c]$, where every non-empty independent set of vertices I has more than $|I|$ neighbors.

It will now be shown that the graph induced on the set H (the union of all MCISs) is König-Egerváry. This fact will be used in the proof of correctness of the parallel algorithm. While the class of König-Egerváry graphs contains all the bipartite graphs (by the König-Egerváry Theorem, [15]) and subgraphs of bipartite graphs are bipartite, it is not true in general that subgraphs of König-Egerváry graphs are König-Egerváry. So it is worth noting that $G[X]$ is König-Egerváry, $G[H]$ is a subgraph of $G[X]$, and $G[H]$ is König-Egerváry.

The following results are required in the proof of Theorem 2.5.

Lemma 2.3. (The Matching Lemma, [11]) *If I is a critical independent set, then there is a matching from $N(I)$ to I .*

Let $\Omega = \Omega(G)$ be the set of maximum independent sets in G . The *core* of a graph G , denoted $core(G)$, is defined to be $\cap \{S : S \in \Omega\}$, namely, the set of vertices which are in every maximum independent set; and $\xi = \xi(G) = |core(G)|$. This notation follows [3].

Theorem 2.4. (Levit & Mandrescu, [13]) *If G is a König-Egerváry graph, then G has a perfect matching if, and only if, $|\cap \{S : S \in \Omega(G)\}| = |\cap \{V(G) - S : S \in \Omega\}|$.*

Theorem 2.5. *If I_c is a maximum critical independent set of a graph G , $X = I_c \cup N(I_c)$, and H is the union of all maximum critical independent sets, then*

1. $H \cup N(H) = X$,
2. $G[H]$ is a König-Egerváry graph,

3. I is a maximum independent set of $G[H]$ if, and only if, I is a MCIS of G and $\alpha'(G) = \alpha(G[H])$,
4. The non-trivial components of $G[H]$ have a perfect matching,
5. If I_0 are the isolated vertices in $G[H]$ then $\alpha(G[H]) = |I_0| + \frac{1}{2}|H \setminus I_0|$.

Proof. Let I_c be a MCIS of a graph G and $X = I_c \cup N(I_c)$. By Theorem 2.1 it follows that, for any MCIS J_c of G , $J_c \cup N(J_c) = X$. Let Ω_c be the set of MCISs of G . Then $H \cup N(H) = \cup\{J_c : J_c \in \Omega_c\} \cup N(\cup\{J_c : J_c \in \Omega_c\}) = [\cup\{J_c : J_c \in \Omega_c\}] \cup [\cup\{N(J_c) : J_c \in \Omega_c\}] = \cup\{J_c \cup N(J_c) : J_c \in \Omega_c\} = X$, proving 1.

Now, $I_c \subseteq H$. Let $H' = H \setminus I_c$. So $n(G[H]) = |I_c| + |H'|$. Furthermore, $\alpha(G[H]) \geq |I_c|$. By construction $H' \subseteq N(I_c)$. By the Matching Lemma there is a matching from $N(I_c)$ into I_c in G . Thus there is a matching from H' into I_c in $G[H]$ and $\mu(G[H]) \geq |H'|$. So $\alpha(G[H]) + \mu(G[H]) \geq |I_c| + |H'| = n(H)$ and, for any graph, $\alpha + \mu \leq n$, it follows that $\alpha(G[H]) + \mu(G[H]) = n(G[H])$, that is, $G[H]$ is König-Egerváry, proving 2. It now follows easily that $\alpha(G[H]) = |I_c|$ and thus that I_c is a maximum independent set of $G[H]$, proving one direction of 3.

Now let I be a maximum independent set of $G[H]$. So I is an independent set in $G[X]$, $|I| \geq |I_c|$, and $\alpha(G[X]) \geq |I|$. It will now be argued that $\alpha(G[X]) = |I_c|$ and, hence, $|I_c| = |I|$. Theorem 2.1 implies that $G[X]$ is König-Egerváry. Then it is not difficult to see that the Matching Lemma implies that $\mu(G[X]) = |N(I_c)|$. Finally, we have $n(G[X]) = \alpha(G[X]) + \mu(G[X]) \geq |I_c| + |N(I_c)| = |X| = n(G[X])$. The claim then follows. Then, since I and I_c are maximum independent sets of $G[H]$, $I \cup N(I) \subseteq H \cup N(H) = X$. $N(I) \subseteq X \setminus I$ and $N(I_c) \subseteq X \setminus I_c$. It is worth noting here that, $N(I)$ is the set of neighbors of I in graph G (as opposed to graph $G[H]$). No use is made in this proof of neighbors of a set of vertices in graph $G[H]$ and no subscripts are ever required for clarity. To continue, it follows that $|N(I)| \leq |X \setminus I|$ and $|N(I_c)| = |X \setminus I_c|$. Since $|X \setminus I| = |X \setminus I_c|$, it follows that $|N(I)| \leq |N(I_c)|$ and, thus, that $|I| - |N(I)| \geq |I| - |N(I_c)|$. But, if $|I| - |N(I)| > |I_c| - |N(I_c)|$, I_c is not a critical independent set, contradicting our assumption. Thus $|I| - |N(I)| = |I_c| - |N(I_c)|$, and I is a critical independent set of G , proving the other direction of 3.

Let I_0 be the set of isolated vertices in $G[H]$. These are contained in any maximum independent set of $G[H]$. Let $I'_c = I_c \setminus I_0$ and $H' = H \setminus I_0$. It is then claimed that $G[H']$ has a perfect matching. Let $v \in H'$. Suppose $v \in \text{core}(G[H'])$, that is, v is in every maximum independent set of $G[H']$. Then v is in every maximum independent set of $G[H]$ and, thus, in every maximum critical independent set of G . But H is the set of vertices in some maximum critical independent set of G . So no vertex in $N(v)$ is in any maximum independent set of $G[H]$, or in any maximum critical independent set of G , which is a contradiction. Thus $|\cap \{S : S \in \Omega(G[H'])\}| = 0$. By similar reasoning it can be shown that $|\cap \{V(G[H']) - S : S \in \Omega(G[H'])\}| = 0$. Thus $|\cap \{S : S \in \Omega(G[H'])\}| = |\cap \{V(G[H']) - S : S \in \Omega(G[H'])\}|$. Theorem 2.4 then implies that $G[H']$ has a perfect matching, proving 4.

It is clear that, since $G[H]$ is König-Egerváry, and $G[H']$ has a perfect matching, $G[H']$ is also König-Egerváry; that is, $\alpha(G[H']) + \mu(G[H']) = n(G[H'])$. Since $n(G[H']) = 2\mu(G[H'])$, it follows that $\alpha(G[H']) = \frac{1}{2}n(G[H']) = \frac{1}{2}|H \setminus I_0|$. Finally $\alpha(G[H]) = |I_0| + \alpha(G[H']) = |I_0| + \frac{1}{2}|H \setminus I_0|$, proving 5. □

3 A parallel MCIS algorithm

The criterion given for testing whether a vertex belongs to a critical independent set begins by passing to a certain bipartite graph. The computational speed of the following algorithm is due to the fact that the independence number of a bipartite graph can be computed in polynomial time.

Definition 3.1. For a graph G , the *bi-double* graph $B(G)$ has vertex set $V \cup V'$, where V' is a copy of V . If $V = \{v_1, v_2, \dots, v_n\}$, let $V' = \{v'_1, v'_2, \dots, v'_n\}$. Then, $(x, y') \in E(B(G))$ if, and only if, $(x, y) \in E(G)$.

The bi-double graph $B(G)$ of G can also be described as $K_2 \square G$, the cartesian product of K_2 and G .

Corollary 3.2. (Larson, [11]) *A graph G contains a non-empty critical independent set if, and only if, there is a vertex $v \in V(G)$ such that $\alpha(B(G)) = \alpha(B(G) - \{v, v'\} - N(\{v, v'\}) + 2$.*

In fact, the proof of this corollary actually shows that a vertex v satisfying the specified condition is in *some* critical set. It is also shown in [11] that any critical independent set can be extended to a MCIS. These results are now combined in a form directly useful in the current context.

Theorem 3.3. (MCIS Criterion) *A vertex v in a graph G is in some MCIS if, and only if, $\alpha(B(G)) = \alpha(B(G) - \{v, v'\} - N(\{v, v'\}) + 2$.*

The following algorithm results in the set of all vertices which are in some maximum critical independent set. Step 4 requires n iterations—but, due to the MCIS Criterion, these n tests can be run independently on n processors. This is where the parallelization takes place.

MCIS subgraph algorithm

1. Construct $B(G)$.
2. $a := \alpha(B(G))$.
3. $H := \emptyset$.
4. For each vertex v in $V(G)$,
 - (a) $t := \alpha(B(G) - \{v, v'\} - N(\{v, v'\}) + 2$.
 - (b) If $t = a$, $H := H \cup \{v\}$.

According to Theorem 3.3 these steps will result in the construction of a set H consisting of all vertices which are in some MCIS. This can be extended in various ways to find the following invariants or sets.

1. Find α' . In order to calculate α' , the remaining step is to identify the trivial and non-trivial components of H . Let I_0 be the isolated vertices in H . Then, by Theorem 2.5, $\alpha'(G) = |I_0| + \frac{1}{2}(|H \setminus I_0|)$.

2. Find X . In order to calculate the decomposition guaranteed by the Independence Decomposition Theorem, it remains to find the neighbors of the vertices in H . Again by Theorem 2.5, $H \cup N(H) = X$. Let $X^c = V(G) \setminus X$. Then $G[X^c]$ will have the property that, for every non-empty independent set J , $|N(J)| > |J|$.
3. Find a MCIS I_c of G . In order to find a MCIS, Theorem 2.5 implies that it suffices to find a maximum independent set I_c in H . Then I_c is a MCIS in G .

Since $B(G)$ is a bipartite graph on $2n$ vertices, calculating a maximum matching of $B(G)$ and, hence, calculating $\alpha(B(G))$ requires $\mathcal{O}(n^{1.5}\sqrt{m/\log n})$ operations, using the algorithm of Alt, et al. [2]. This algorithm will be run once and then a second time independently on each of n processors. So the total running time is still $\mathcal{O}(n^{1.5}\sqrt{m/\log n})$.

If M is a matching in a graph G and w is a vertex incident to an edge in M , let w' be the vertex matched to w under M . The new algorithm can now be stated. The parallelism occurs in step 1.

The parallel MCIS algorithm

1. Find H .
2. Find the set I_0 of isolated vertices in $G[H]$. $H_0 := I_0$.
3. If $H \setminus H_0 = \emptyset$, $I := I_0$. STOP.
4. Find a maximum matching M of $G[H]$.
5. Let $w \in H \setminus H_0$.
6. $N_1 := N(I_0 \cup \{w\})$, $M_1 := \{v' : v \in N_1\}$, $I_1 := I_0 \cup M_1$, $H_1 := I_1 \cup N_1$ ($=H_0 \cup N_1 \cup M_1$).
7. $i := 1$.
8. While $H \setminus H_i \neq \emptyset$:
 - (a) i. If $H_i \neq H_{i-1}$: $N_{i+1} := N(I_i)$.
 - ii. Else if $H_i = H_{i-1}$:
 - A. Let $w \in H \setminus H_i$.
 - B. $N_{i+1} := N(I_i \cup \{w\})$,
 - (b) $M_{i+1} := \{v' : v \in N_{i+1}\}$, $I_{i+1} := I_i \cup M_{i+1}$. $H_{i+1} := I_{i+1} \cup N_{i+1}$, $i := i + 1$.
9. $I := I_i$.

Theorem 3.4. *If G is a graph then the set I produced by the Parallel MCIS algorithm is a maximum critical independent set of G .*

Proof. Let G be a graph, H be the set of vertices in some maximum critical independent set of G , and M be the maximum matching of $G[H]$ produced by the Parallel MCIS algorithm. Theorem 2.5 implies that $\alpha'(G) = \alpha(G[H])$. Thus it is enough to show that the set I produced by this algorithm is a maximum independent set of $G[H]$.

Let I_0 be the set of isolated vertices in $G[H]$ and $H' = H \setminus I_0$. It was shown that $G[H]$ is a König-Egerváry graph whose non-trivial components have perfect matchings. $G[H']$ is the union of the non-trivial components. So M is a perfect matching of $G[H']$.

The algorithm will first be described for the convenience of the reader. The first step is to identify the isolated vertices. These can be extended to a maximum independent set of $G[H]$. Then choose any vertex w among the remaining vertices. By the definition of the set H , there is a MCIS and, by Theorem 2.5, this is a maximum independent set of $G[H]$. So there is maximum independent set I of $G[H]$ which contains w . The neighbors of this vertex cannot be in I but each of these vertices is incident to some edge in the perfect matching M of $G[H]$ and, since one vertex from every edge of M must be in I , it follows that the vertices matched to $N(w)$ under M must be in I . In general, having constructed an independent set J , the neighbors of J cannot be in the maximum independent set but, since one vertex from every edge in M must be in the maximum independent set, the vertices matched to $N(J)$ under M must be in the set. If at some point there are no new vertices in $N(J)$, but the vertices in the graph have not been used up, an arbitrary vertex can be selected from the remaining vertices, added to the independent set, and the process continued.

The set I_0 is an independent set, $H_0 = I_0$, and there is a maximum independent set of $G[H]$ containing I_0 . Assume that after the $(k - 1)$ th iteration of the while loop, I_k is an independent set and there is a maximum independent set of $G[H]$ containing I_k . It will be shown that I_{k+1} is an independent set and there is a maximum independent set of $G[H]$ which contains I_{k+1} .

If $H \setminus H_k = \emptyset$ after the $(k - 1)$ th iteration of the while loop, then $H = H_k$ and I is a maximum independent set of $G[H]$. Assume then that $H \setminus H_k \neq \emptyset$ after the $(k - 1)$ th iteration of the while loop. H_k is formed by (possibly) adding vertices to H_{k-1} , namely, $N(I_{k-1}) \setminus H_{k-1}$ together with the vertices matched to these under M . Either $H_k \neq H_{k-1}$ or $H_k = H_{k-1}$. Note that, in either case, by construction, $N_k \subseteq N_{k+1}$, $M_k \subseteq M_{k+1}$, $I_k \subseteq I_{k+1}$, and $H_k \subseteq H_{k+1}$.

In the first case, $H_k \setminus H_{k-1} \neq \emptyset$. H_k is formed by adding the vertices $N_k \setminus N_{k-1}$ and their neighbors $M_k \setminus M_{k-1} = I_k \setminus I_{k-1}$ under the matching M . The vertices $M_k \setminus M_{k-1}$ may or may not have neighbors outside of H_k . $N_{k+1} = N(I_k)$, $M_{k+1} = \{v' : v \in N_{k+1}\}$, $I_{k+1} = I_k \cup M_{k+1}$, and $H_{i+1} = I_{i+1} \cup N_{i+1}$. By assumption I_k is an independent set and there is a maximum independent set of $G[H]$ containing I_k . The vertices in I_{k+1} are the vertices in I_k together with the vertices matched to the neighbors of I_k under M . Let I be a maximum independent set of $G[H]$ containing I_k . It cannot contain any neighbor of I_k . Since any maximum independent set I of $G[H]$ must contain one vertex from each edge of M , I must contain the vertices matched to $N(I_k)$ under M . Thus I_{k+1} is an independent set and it can be extended to a maximum independent set of $G[H]$.

In the case where $H_k = H_{k-1}$, the k th step in the while loop of the algorithm (step 8) works as follows: A vertex w is selected from from $H \setminus H_k$. Since I_k is independent and $w \notin N(w)$, $I_{k+1} = I_k \cup \{w\}$ is an independent set. By assumption there is a MCIS containing w and, following Theorem 2.5, there is a maximum independent set I containing w . Each edge in M must be incident to some vertex in I . Let $I' = (I \setminus H_k) \cup I_k$. It remains to be shown that I' is a maximum independent set of $G[H]$. Since $H_k = I_k \cup N(I_k)$ it follows that I' is an independent set. It is now enough to show that, for every edge xy in M , either x or y is in I' . Either x or y is in I . Assume $x \in I$. If $x \notin I'$ then $x \in H_k$. But then by the construction of H_k , y is matched to x under M and y is also in H_k . But I_k is a maximum independent set in $G[H_k]$. So either x or y must be in I_k and, thus I' . So I_{k+1} is contained in a maximum independent set.

□

The MCIS Subgraph algorithm, which produces H , requires $\mathcal{O}(n^{1.5}\sqrt{m/\log n})$ operations. Finding a maximum matching of $G[H]$ requires the same order or of operations. The remaining steps only require finding the neighbors of sets of vertices. So the total running time of Parallel MCIS Algorithm is $\mathcal{O}(n^{1.5}\sqrt{m/\log n})$.

4 Acknowledgment and future research

We would like to thank an anonymous referee who pointed us to state-of-the-art references on maximum matching algorithms (including [8, 9, 10]). The referee notes that algorithm performance depends on edge density: some algorithms perform better for sparse graphs while others perform better for dense graphs.

The referee also suggests that future investigation of the time complexity of the algorithm presented above would be of interest in the case that there are n^2 or n^3 processors. The presented algorithm requires use of n processors; the problem naturally breaks into n cases following Theorem 3.3. We do not know of a way to break the problem down further.

References

- [1] A. Ageev, On finding critical independent and vertex sets, *SIAM J. Discrete Mathematics* **7** (1994), 293–295.
- [2] H. Alt, N. Blum, K. Melhorn and M. Paul, Computing a maximum cardinality matching in a bipartite graph in time $\mathcal{O}(n^{1.5}\sqrt{m/\log n})$, *Information Processing Letters* **37** (1991), 237–240.
- [3] E. Boros, M. Golumbic and V. Levit, On the Number of Vertices belonging to all Maximum Stable Sets of a Graph, *Discrete Applied Mathematics* **124** (2002), 17–25.
- [4] S. Butenko and S. Trukhanov, Using Critical Sets to Solve the Maximum Independent Set Problem, *Operations Research Letters* **35** (2007), 519–524.
- [5] E. DeLaViña, Written on the Wall II, Conjectures of Graffiti.pc, <http://cms.uhd.edu/faculty/delavinae/research/wowII/index.htm>.
- [6] R. W. Deming, Independence Numbers of Graphs—an Extension of the Koenig-Egervary Theorem, *Discrete Mathematics* **27** (1979), 23–33.
- [7] M. Garey and D. Johnson, *Computers and Intractability*, W. H. Freeman and Company, New York, 1979.
- [8] N. J. A. Harvey, *Matchings, Matroids and Submodular Functions*, Massachusetts Institute of Technology Ph.D. thesis, 2008.
- [9] A. Jindal, G. Kochar and M. Pal, Maximum Matchings via Glauber Dynamics, Arxiv preprint arXiv:1107.2482, 2011.
- [10] K. Kaya, J. Langguth, F. Manne and B. Uc, Experiments on Push-Relabel-based Maximum Cardinality Matching Algorithms for Bipartite Graphs, Technical Report TR/PA/11/33, CER-FACS, Toulouse, France, 2011.
- [11] C. E. Larson, A Note on Critical Independence Reductions, *Bulletin of the Institute of Combinatorics and its Applications* **51** (2007), 34–46.
- [12] C. E. Larson, Critical Independent Sets and an Independence Decomposition Theorem, *European Journal of Combinatorics* **32** (2011), 294–300.
- [13] V. E. Levit and E. Mandrescu, On α^+ -stable König-Egervary graphs, *Discrete Mathematics* **263** (2003) 179–190.

- [14] V. E. Levit and E. Mandrescu, Critical Independent Sets and König-Egerváry Graphs, *Graphs and Combinatorics* **28** (2012), 243–250.
- [15] L. Lovasz, M. D. Plummer, *Matching Theory*, North Holland, Amsterdam, 1986.
- [16] F. Sterboul, A characterization of the graphs in which the transversal number equals the matching number, *Journal of Combinatorial Theory. Series B* **27** (1979), 228–229.
- [17] C.-Q. Zhang, Finding critical independent sets and critical vertex subsets are polynomial problems, *SIAM J. Discrete Mathematics* **3** (1990), 431–438.