Maximum Cardinality Matching

Christine Rizkallah

March 17, 2025

Abstract

A *matching* in a graph G is a subset M of the edges of G such that no two share an endpoint. A matching has maximum cardinality if its cardinality is at least as large as that of any other matching. An *odd-set cover* OSC of a graph G is a labeling of the nodes of G with integers such that every edge of G is either incident to a node labeled 1 or connects two nodes labeled with the same number $i \geq 2$.

Theorem 1 (Edmonds [2])**.** Let M be a matching in a graph G and let OSC be an odd-set cover of G. For any $i \geq 0$, let n_i be the number of nodes labeled i. If

$$|M| = n_1 + \sum_{i \geq 2} \lfloor n_i/2 \rfloor$$

then M is a maximum cardinality matching.

We provide an Isabelle proof of Edmonds theorem. For an explanation of the proof see [1].

Contents

theory *Matching*
imports *Main*
begin

type-synonym *label* = *nat*

1 Definitions

definition $\mathit{finite\text{-}graph} :: {'v}\ set => ({'v} * {'v})\ set \Rightarrow bool$ **where**
$\mathit{finite\text{-}graph}\ V\ E = (\mathit{finite}\ V \wedge \mathit{finite}\ E \wedge$
$(\forall\ e \in E.\ fst\ e \in V \wedge snd\ e \in V \wedge fst\ e \ {}^{\sim}{=}\ snd\ e))$

definition $\mathit{degree} :: ({'v} * {'v})\ set \Rightarrow {'v} \Rightarrow nat$ **where**
$\mathit{degree}\ E\ v = card\ \{e \in E.\ fst\ e = v \vee snd\ e = v\}$

definition $\mathit{edge\text{-}as\text{-}set} :: ({'v} * {'v}) \Rightarrow {'v}\ set$ **where**
$\mathit{edge\text{-}as\text{-}set}\ e = \{fst\ e,\ snd\ e\}$

definition $N :: {'v}\ set \Rightarrow ({'v} \Rightarrow label) \Rightarrow nat \Rightarrow nat$ **where**
$N\ V\ L\ i = card\ \{v \in V.\ L\ v = i\}$

definition $\mathit{weight} :: label\ set \Rightarrow (label \Rightarrow nat) \Rightarrow nat$ **where**
$\mathit{weight}\ LV\ f = f\ 1 + (\sum i{\in}LV.\ (f\ i)\ div\ 2)$

definition $OSC :: ({'v} \Rightarrow label) \Rightarrow ({'v} * {'v})\ set \Rightarrow bool$ **where**
$OSC\ L\ E = (\forall e \in E.\ L\ (fst\ e) = 1 \vee L\ (snd\ e) = 1 \vee$
$L\ (fst\ e) = L\ (snd\ e) \wedge L\ (fst\ e) > 1)$

definition $\mathit{disjoint\text{-}edges} :: ({'v} * {'v}) \Rightarrow ({'v} * {'v}) \Rightarrow bool$ **where**
$\mathit{disjoint\text{-}edges}\ e1\ e2 = (fst\ e1 \neq fst\ e2 \wedge fst\ e1 \neq snd\ e2 \wedge$
$snd\ e1 \neq fst\ e2 \wedge snd\ e1 \neq snd\ e2)$

definition $\mathit{matching} :: {'v}\ set \Rightarrow ({'v} * {'v})\ set \Rightarrow ({'v} * {'v})\ set \Rightarrow bool$ **where**
$\mathit{matching}\ V\ E\ M = (M \subseteq E \wedge \mathit{finite\text{-}graph}\ V\ E \wedge$
$(\forall e1 \in M.\ \forall\ e2 \in M.\ e1 \neq e2 \longrightarrow \mathit{disjoint\text{-}edges}\ e1\ e2))$

definition $\mathit{matching\text{-}i} :: nat \Rightarrow {'v}\ set \Rightarrow ({'v} * {'v})\ set \Rightarrow ({'v} * {'v})\ set \Rightarrow$
$({'v} \Rightarrow label) \Rightarrow ({'v} * {'v})\ set$ **where**
$\mathit{matching\text{-}i}\ i\ V\ E\ M\ L = \{e \in M.\ i{=}1 \wedge (L\ (fst\ e) = i \vee L\ (snd\ e) = i)$
$\vee\ i{>}1 \wedge L\ (fst\ e) = i \wedge L\ (snd\ e) = i\}$

definition $V\text{-}i :: nat \Rightarrow {'v}\ set \Rightarrow ({'v} * {'v})\ set \Rightarrow ({'v} * {'v})\ set \Rightarrow$
$({'v} \Rightarrow label) \Rightarrow {'v}\ set$ **where**
$V\text{-}i\ i\ V\ E\ M\ L = \bigcup\ (\mathit{edge\text{-}as\text{-}set}\ `\ \mathit{matching\text{-}i}\ i\ V\ E\ M\ L)$

definition $\mathit{endpoint\text{-}inV} :: {'v}\ set \Rightarrow ({'v} * {'v}) \Rightarrow {'v}$ **where**
$\mathit{endpoint\text{-}inV}\ V\ e = (if\ fst\ e \in V\ then\ fst\ e\ else\ snd\ e)$

definition $\mathit{relevant\text{-}endpoint} :: ({'v} \Rightarrow label) \Rightarrow {'v}\ set \Rightarrow$
$({'v} * {'v}) \Rightarrow {'v}$ **where**
$\mathit{relevant\text{-}endpoint}\ L\ V\ e = (if\ L\ (fst\ e) = 1\ then\ fst\ e\ else\ snd\ e)$

2 Lemmas

lemma *definition-of-range*:

endpoint-inV V1 ‘ matching-i 1 V E M L =
{ *v*. ∃ *e* ∈ *matching-i 1 V E M L*. *endpoint-inV V1 e* = *v* } ⟨*proof*⟩

lemma *matching-i-edges-as-sets*:
edge-as-set ‘ matching-i i V E M L =
{ *e1*. ∃ (*u*, *v*) ∈ *matching-i i V E M L*. *edge-as-set* (*u*, *v*) = *e1*} ⟨*proof*⟩

lemma *matching-disjointness*:
assumes *matching V E M*
assumes *e1* ∈ *M*
assumes *e2* ∈ *M*
assumes *e1* ≠ *e2*
shows *edge-as-set e1* ∩ *edge-as-set e2* = {}
⟨*proof*⟩

lemma *expand-set-containment*:
assumes *matching V E M*
assumes *e* ∈ *M*
shows *e* ∈ *E*
⟨*proof*⟩

theorem *injectivity*:
assumes *is-osc*: *OSC L E*
assumes *is-m*: *matching V E M*
assumes *e1-in-M1*: *e1* ∈ *matching-i 1 V E M L*
and *e2-in-M1*: *e2* ∈ *matching-i 1 V E M L*
assumes *diff*: (*e1* ≠ *e2*)
shows *endpoint-inV* {*v* ∈ *V*. *L v* = *1*} *e1* ≠ *endpoint-inV* {*v* ∈ *V*. *L v* = *1*}
e2
⟨*proof*⟩

2.1 |*M1*| ≤ *n1*

lemma *card-M1-le-NVL1*:
assumes *matching V E M*
assumes *OSC L E*
shows *card* (*matching-i 1 V E M L*) ≤ (*N V L 1*)
⟨*proof*⟩

lemma *edge-as-set-inj-on-Mi*:
assumes *matching V E M*
shows *inj-on edge-as-set* (*matching-i i V E M L*)
⟨*proof*⟩

lemma *card-Mi-eq-card-edge-as-set-Mi*:
assumes *matching V E M*
shows *card* (*matching-i i V E M L*) = *card* (*edge-as-set‘ matching-i i V E M L*)
(**is** *card ?Mi* = *card* (*?f* ‘ -))
⟨*proof*⟩

lemma *card-edge-as-set-Mi-twice-card-partitions*:
assumes *OSC L E* ∧ *matching V E M* ∧ *i* > *1*
shows *2* ∗ *card* (*edge-as-set*‘*matching-i i V E M L*)
= *card* (*V-i i V E M L*) (**is** *2* ∗ *card ?C* = *card ?Vi*)
⟨*proof*⟩

lemma *card-Mi-twice-card-Vi*:
assumes *OSC L E* ∧ *matching V E M* ∧ *i* > *1*
shows *2* ∗ *card* (*matching-i i V E M L*) = *card* (*V-i i V E M L*)
⟨*proof*⟩

lemma *card-Mi-le-floor-div-2-Vi*:
assumes *OSC L E* ∧ *matching V E M* ∧ *i* > *1*
shows *card* (*matching-i i V E M L*) ≤ (*card* (*V-i i V E M L*)) *div 2*
⟨*proof*⟩

lemma *card-Vi-le-NVLi*:
assumes *i*>*1* ∧ *matching V E M*
shows *card* (*V-i i V E M L*) ≤ *N V L i*
⟨*proof*⟩

2.2 $|Mi| \leq \lfloor ni/2 \rfloor$

lemma *card-Mi-le-floor-div-2-NVLi*:
assumes *OSC L E* ∧ *matching V E M* ∧ *i* > *1*
shows *card* (*matching-i i V E M L*) ≤ (*N V L i*) *div 2*
⟨*proof*⟩

2.3 $|M| \leq \sum |Mi|$

lemma *card-M-le-sum-card-Mi*:
assumes *matching V E M* **and** *OSC L E*
shows *card M* ≤ (∑ *i* ∈ *L*‘*V*. *card* (*matching-i i V E M L*))
(**is** *card* - ≤ *?CardMi*)
⟨*proof*⟩

theorem *card-M-le-weight-NVLi*:
assumes *matching V E M* **and** *OSC L E*
shows *card M* ≤ *weight* {*i* ∈ *L* ‘ *V*. *i* > *1*} (*N V L*) (**is** - ≤ *?W*)
⟨*proof*⟩

3 Final Theorem

The following theorem is due to Edmond [2]:

theorem *maximum-cardinality-matching*:
assumes *matching V E M* **and** *OSC L E*
and *card M* = *weight* {*i* ∈ *L* ‘ *V*. *i* > *1*} (*N V L*)

and *matching* V E M'
shows *card* $M' \leq$ *card* M
⟨*proof*⟩

The widely used algorithmic library LEDA has a certifying algorithm for maximum cardinality matching. This Isabelle proof is part of the work done to verify the checker of this certifying algorithm. For more information see [1].

end

References

[1] E. Alkassar, S. Böhme, K. Mehlhorn, and C. Rizkallah. Verification of certifying computations. In *Proceedings of the 23rd International Conference on Computer Aided Verfication (CAV2011)*, Cliff Lodge, Snowbird, Utah, USA, 2011. To Appear.

[2] J. Edmonds. Maximum matching and a polyhedron with 0,1 - vertices. *Journal of Research of the National Bureau of Standards*, 69B:125–130, 1965.