



HAL
open science

Packing non-returning A-paths algorithmically

Gyula Pap

► **To cite this version:**

Gyula Pap. Packing non-returning A-paths algorithmically. 2005 European Conference on Combinatorics, Graph Theory and Applications (EuroComb '05), 2005, Berlin, Germany. pp.139-144, 10.46298/dmtcs.3399 . hal-01184355

HAL Id: hal-01184355

<https://inria.hal.science/hal-01184355v1>

Submitted on 14 Aug 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Packing non-returning A -paths algorithmically

Gyula Pap¹ †

¹ Dept. of Operations Research, Eötvös University, Pázmány P. s. 1/C, Budapest, Hungary H-1117.

In this paper we present an algorithmic approach to packing A -paths. It is regarded as a generalization of Edmonds' matching algorithm, however there is the significant difference that here we do not build up any kind of alternating tree. Instead we use the so-called 3-way lemma, which either provides augmentation, or a dual, or a subgraph which can be used for contraction. The method works in the general setting of packing non-returning A -paths. It also implies an ear-decomposition of criticals, as a generalization of the odd ear-decomposition of factor-critical graph.

Keywords: A -paths, matching

1 Introduction

The paper is devoted to the problem of packing fully node-disjoint non-returning A -paths in a graph $G = (V, E)$. Given a graph and a subset $A \subseteq V$, a path is said to be an A -path if its ends are two distinct nodes in A . Packing fully node-disjoint A -paths reduces to maximum matching in an auxiliary graph, see T. Gallai (3). The special case $A = V$ is in fact equivalent to maximum matching. W. Mader considered a more difficult problem. We are given a subset $A \subseteq V$ with a partition \mathcal{A} . An A -path is called an \mathcal{A} -path if its ends are in two distinct members of \mathcal{A} . Mader (5) gave a min-max formula for the maximum number of fully node-disjoint \mathcal{A} -paths. A polynomial time algorithm to find these paths was given by L. Lovász using his matroid parity apparatus. Matroid parity is still a challenging topic in combinatorial optimization. If a problem turns out to be an instance for matroid parity, this does not necessarily imply a polynomial time algorithm or a good characterization. Lovász disentangled some technical details to construct an algorithm, see (4). Later, A. Schrijver gave a funny reduction to linear matroid parity – which by itself also implies an algorithm. It was a challenge to construct directly an algorithm for packing \mathcal{A} -paths. Such an algorithm was given by Chudnovsky et al. (2). They in fact work with the concept of non-zero A -paths, which is a generalization of \mathcal{A} -paths, see also (1). The main goal of this paper is to construct an algorithm which presents the “dual” in a more structured form. Our method implies an ear-decomposition of “criticals” – this generalizes the ear-decomposition of factor-critical graphs.

Maximum matching is a special case of the problem discussed in this paper, let us briefly sketch how the method works for maximum matching. For a given matching $M \subseteq E$ in G , we call an odd cycle $C \subseteq E$ an M -alternating odd cycle if $|C \cap M| = (|C| - 1)/2$ and C is incident to an M -exposed node. The following lemma can be proved directly, a proof “on the level of bipartite matching” can be given. In

† Research is supported by OTKA grants T 037547 and TS 049788, by European MCRTN Adonet, Contract Grant No. 504438 and by the Egerváry Research Group of the Hungarian Academy of Sciences. e-mail: gyuszk@cs.elte.hu

fact, Edmonds' alternating forests provide an alternative proof of this lemma. Our crucial observation is that a matching algorithm can be constructed by only using the below lemma as a black box. This black box is regarded as a compact formulation of some consequences of alternating forests. However, one can also give a short, inductive proof without alternating forests.

Lemma 1.1 (3-Way Lemma for Matching) *Given an undirected graph G with a matching M , then at least one of the following alternatives holds:*

1. *There is a matching N with $|N| = |M| + 1$.*
2. *There is a matching N with $|N| = |M|$ and an N -alternating odd cycle in G .*
3. *There is a vector $c \in \{0, 1, 2\}^V$ such that the weight of any edge is at least 2, and the sum of its entries is exactly $2|M|$.*

This lemma allows us to interpret of Edmonds' algorithm as follows. Consider a matching M in graph G , try Lemma 1.1. Alternative 1 gives an augmentation, alternative 3 verifies optimality. Alternative 2 provides an odd cycle for contraction. Contraction of an alternating odd cycle has the property that augmentation, or a Berge-Tutte-dual in G/C can be expanded to G .

2 Packings in p-graphs — Definitions

The most important notion in this paper is a **permutation labeled graph** or **p-graph**, for short. A p-graph comes in the form of G, A, ω, π , where G is a graph, A is a set of nodes, π are edge-labels. This notion provides a generalization of some well-known packing problems – matching, node-disjoint A -paths, non-zero A -paths. The motivation for this version is that important reduction principles used by our algorithm stay within the concept of a p-graph, but does not stay within well-known previous concepts. The precise definition of a p-graph is formulated as follows.

Let $G = (V, E)$ be an undirected graph with node-set V , edge-set E with a reference orientation. Let $A \subseteq V$ be a fixed set of **terminals**. Let Ω be an arbitrary **set of “potentials”** and let **jj, JJ** be called **Jolly Joker** (some imaginary labels). Let $\omega : A \rightarrow \Omega$ define the **potential of origin** for the terminals. Let $\pi : E \rightarrow S(\Omega) \cup \{\mathbf{JJ}\}$ where $S(\Omega)$ is the set of all permutations of Ω . For an edge $ab = e \in E$, let $\pi(e, a) := \pi(e)$ and $\pi(e, b) := \pi^{-1}(e)$ be the **mapping of potential** on edge ab . (We use \circ for the composition of permutations. We define $\mathbf{JJ}^{-1} := \mathbf{JJ} \circ \pi := \pi \circ \mathbf{JJ} := \mathbf{JJ}$ and $\mathbf{JJ}(\omega) := \pi(\mathbf{jj}) := \mathbf{jj}$ for any $\pi \in S(\Omega) \cup \{\mathbf{JJ}\}$ and for any $\omega \in \Omega \cup \{\mathbf{jj}\}$.) A **walk** in G is a sequence of nodes and edges, say $W = (v_0, e_0, v_1, e_1, \dots, e_{k-1}, v_k)$ where $e_i = v_i v_{i+1}$ or $e_i = v_{i+1} v_i$ for all $0 \leq i \leq k-1$. W is called an **A -walk** in G if $v_0, v_k \in A$ and $v_j \notin A$ (for $j \neq 0, k$). $\chi_W \in \mathbb{N}^V$ denotes the **traversing multiplicity vector** of walk W , defined by $\chi_W(v) := |\{j : v_j = v\}|$. A walk W is called a **path** if $\chi_W \leq \mathbf{1}$. We will usually use letters P, R for paths. For an A -walk let $\pi(W) := \pi(e_0, v_0) \circ \pi(e_1, v_1) \circ \dots \circ \pi(e_{k-1}, v_{k-1})$ define the **mapping of potentials on W** . W is called **non-returning** if $\pi(W)(\omega(v_0)) \neq \omega(v_k)$. (Hence, an empty A -walk (having a single node and no edge) is not considered to be non-returning. Notice, if W traverses any edge with label **JJ**, then W is non-returning.) A family \mathcal{P} of fully node-disjoint non-returning A -paths is called a **packing**. $\nu = \nu(G) = \nu(G, A, \omega, \pi)$ denotes the **maximum cardinality of a packing**. Also, a “node-capacited packing problem” can be defined. Consider a function $b \in \mathbb{N}^V$ of **node capacities**. A family \mathcal{W} of A -walks (we allow walks to be taken multiply) is called a **b -packing** if $\sum_{W \in \mathcal{W}} \chi_W \leq b$. Let $\nu_b = \nu_b(G) = \nu_b(G, A, \omega, \pi)$ denotes the maximum cardinality of a b -packing. $b = \mathbf{1}$ defines packings, $b = \mathbf{2}$ defines 2-packings.

3 Min-max Theorems for packings

For a set $F \subseteq E$ of edges, let $A^F := A \cup V(F)$. F is called **A -balanced** if ω can be extended to a function $\omega^F : A^F \rightarrow \Omega$ s.t. each edge $ab \in F$ is ω^F -**balanced** – i.e. $\pi(ab, a)(\omega^F(a)) = \omega^F(b)$. Let $c_{\text{odd}}(G, A)$ be the number of components in G having an odd number of nodes in A – these will be called **odd components of G, A** . Let $c_1(G, A)$ be the number of nodes in A which are isolated nodes of G .

Theorem 3.1 *In a p -graph the maximum cardinality of a packing is determined by*

$$\nu(G, A, \omega, \pi) = \min_{F, X} |X| + \frac{1}{2} (|A^F - X| - c_{\text{odd}}(G - F - X, A^F - X)) , \quad (1)$$

where the minimum is taken over an A -balanced edge-set F and a set $X \subseteq V$.

Theorem 3.2 *In a p -graph the maximum cardinality of a 2-packing is determined by*

$$\nu_2(G, A, \omega, \pi) = \min_{F, X} 2|X| + |A^F - X| - c_1(G - F - X, A^F - X) , \quad (2)$$

where the minimum is taken over an A -balanced edge-set F and a set $X \subseteq V$.

In Theorem 3.2 we do not count odd components to determine a maximum 2-packing, this indicates that 2-packings are simpler than packings. A similar relation there is between matchings and 2-matchings, the latter admitting a reduction to bipartite matching, Kőnig's Theorem. The following theorem is in fact a reformulation of Theorem 3.2, here we formulate a Kőnig-type condition for 2-packings.

Theorem 3.3 *In a p -graph the maximum cardinality of a 2-packing is determined by*

$$\nu_2(G, A, \omega, \pi) = \min \|c\| , \quad (3)$$

where $\|c\| := \sum_{v \in V} c(v)$ and the minimum is taken over **2-covers** c , i.e. vectors $c \in \{0, 1, 2\}^V$ such that $c \cdot \chi_W \geq 2$ for any non-returning A -walk.

4 Contraction of dragons

A path P is called a **half- A -path** if it starts in a terminal $s \in A$, ends in a node $t \in V$ and $V(P) \cap A = \{s\}$. We say P **ends in t with potential** $\pi(P)(\omega(s))$. Consider a node $v \in V$ and a potential $\omega_0 \in \Omega \cup \{\mathbf{jj}\}$. We say a node v is **ω_0 -reachable** (or ω_0 is reachable at v), if there is a pair \mathcal{P}, P_v such that P_v is a half- A -path ending in v with ω_0 , and \mathcal{P} is a packing of ν non-returning A -paths each of which is fully node-disjoint from P_v . We say a node is **reachable** if it is ω_0 -reachable for some $\omega_0 \in \Omega \cup \{\mathbf{jj}\}$. v is called **uniquely reachable** if it is ω_0 -reachable only with a single element $\omega_0 \neq \mathbf{jj}$. Otherwise – if v is **jj-reachable** or there are at least two different elements of Ω which are reachable at v , then v is called **multiply reachable**. The definition implies that a reachable terminal is uniquely reachable. We call a p -graph G a **dragon** if $|A| = 2\nu + 1$ and every node is reachable. A p -graph is called **critical** if it is a dragon such that every non-terminal is multiply reachable. (The notion of criticals is analogue to the notion used in (1). The notion of dragons should be considered as a weak version of criticality.) Let us use the expression **odd cycle** for p -graphs s.t. $G = (V, E)$ is an odd cycle, $A = V$, and all the edges in E give one-edge non-returning A -walks (which are in fact non-returning A -paths except for 1-edge odd cycles). A p -graph with $V = \{a, b\}$, $E = \{ab\}$, $A = \{a\}$ is called a **rod**.

Claim 4.1 *Odd cycles and rods are dragons.* □

A crucial lemma is the following, saying that the min-max formula holds for dragons.

Lemma 4.2 (A dragon has a special dual) *Suppose a G is a dragon with exactly its nodes in V_1 being uniquely reachable, say $v \in V_1$ is $\omega'(v)$ -reachable. Let $F := \{e \in E[V_1] : e \text{ is } \omega'\text{-balanced}\}$. Then $2\nu = |V_1| - c(G - F, V_1)$.*

The notion “reachability” is in fact motivated by the goal to define the contraction of dragon subgraphs.

Definition 4.3 (Contraction of a dragon) *Consider a set $Z \subseteq V$ such that $G[Z]$ is dragon. We define the contracted p-graph on G/Z as follows. Let Z_1 be the uniquely reachable nodes in $G[Z]$, say $a \in Z_1$ is ω_a -reachable. Let $A/Z := A - Z + \{Z\}$. Let $\Omega' := \Omega + \bullet$ for some new element $\bullet \notin \Omega$. Let $\omega_Z(s) := \omega(s)$ for all $s \in A/Z - \{Z\}$, and let $\omega_Z(\{Z\}) := \bullet$. We define $\pi_Z(e)$ by the following case splitting. If e is disjoint from Z , then we define $\pi_Z(e)$ by extending $\pi(e)$ to Ω' by mapping \bullet to \bullet . For an edge ab with $a \in Z_1$, $b \notin Z$ we label its image $\{Z\}b$ s.t. $\pi_Z(\{Z\}b)(\{Z\}) = \pi(ab)(\omega_a)$. For an edge ab with $a \in Z - Z_1$, $b \notin Z$ we define let $\pi_Z(\{Z\}b) := \mathbf{JJ}$.*

We define the **contraction of a node-disjoint family \mathcal{Z} of dragons** $G/\mathcal{Z}, A/\mathcal{Z}, \omega_{\mathcal{Z}}, \pi_{\mathcal{Z}}$ by contracting the dragons in \mathcal{Z} one-by-one. By definition, a contraction has the following properties.

Claim 4.4 (Expansion of a packing) *From any packing in G/\mathcal{Z} one can construct a packing in G which exposes the same number of terminals.*

Claim 4.5 (Pre-image of a dragon) *The pre-image of a dragon Z_1 in G/\mathcal{Z} is dragon. (Thus, $\mathcal{Z}/Z_1 := \{Z : Z \in \mathcal{Z}, \{Z\} \notin Z_1\} \cup \{\text{the pre-image of } Z_1\}$ is a finer node-disjoint family of dragons.)*

5 The 3-Way Lemma and the algorithm

Our main tool in the algorithm is the 3-Way Lemma for packings. Consider a packing \mathcal{P} in G and a dragon Z in G . We say \mathcal{P} is **equipped with Z** if \mathcal{P} consists of some paths disjoint from $V(Z)$ and exactly $\nu(G[Z]) = (|A \cap V(Z)| - 1)/2$ paths inside Z .

Lemma 5.1 (The 3-way Lemma) *Consider a p-graph with a packing \mathcal{P} . Then at least one of the following alternatives holds:*

1. *There is a packing \mathcal{R} with $|\mathcal{R}| = |\mathcal{P}| + 1$.*
2. *There is a packing \mathcal{R} s.t. $|\mathcal{R}| = |\mathcal{P}|$, and is equipped with a rod or an odd cycle.*
3. *There is a 2-cover c such that $2|\mathcal{P}| = \|c\|$. (I.e. a verifying 2-cover for $2 \times \mathcal{P}$)*

The 3-Way Lemma is applied sequentially in the algorithm to construct sequences of contractions. A **sequence of contractions** is a sequence $(\mathcal{Z}_1, G_1, \mathcal{P}_1, \mathcal{R}_1, S_1), \dots, (\mathcal{Z}_m, G_m, \mathcal{P}_m, \mathcal{R}_m, S_m), (\mathcal{Z}_{m+1}, G_{m+1}, \mathcal{P}_{m+1})$ with $m \geq 0$, and the following properties. $\mathcal{Z}_0 = \emptyset$, and \mathcal{Z}_i is a node-disjoint family of dragons in G . $G_i = (V_i, E_i) := G/\mathcal{Z}_i$. $G_i[S_i]$ is an odd cycle or a rod, where $S_i \subseteq V_i$. \mathcal{R}_i is a packing in G_i which is equipped with S_i . $\mathcal{P}_{i+1} := \mathcal{R}_i/S_i$, $\mathcal{Z}_{i+1} := \mathcal{Z}_i/S_i$ for $i = 1, \dots, m$. Each $\mathcal{P}_i, \mathcal{R}_i$ leaves the same number of terminals uncovered.

The proof of Theorem 3.1 and the algorithm relies on the following key observation, which provides a tool to construct a verifying pair. It says that from a 2-packing verification in a contraction we can construct a packing verification in the original p-graph.

Lemma 5.2 (Constructing a verifying pair) *Suppose we have a sequence of contractions, and a 2-cover c in G_{m+1} with $2|\mathcal{P}_{m+1}| = |c|$. Then for all i , \mathcal{P}_i is a maximum packing in G_i and one can construct a verifying pair for \mathcal{P}_i .*

Now we are in position to sketch the algorithm. Our algorithm has an input of a p -graph G and a packing \mathcal{P} . The output is either a larger packing, or a verifying pair for \mathcal{P} . The algorithm starts off with initiating the trivial sequence of contractions, $m = 0$. In a general step, apply Lemma 5.1 to $G_{m+1}, \mathcal{P}_{m+1}$! If alternative 1 holds, then by Claim 4.4 one can construct a packing in G larger than \mathcal{P} . If alternative 2 holds, then by Claim 4.5 one can construct a longer sequence of contractions. If alternative 3 holds, then by Claim 5.2 \mathcal{P} is maximum, and a verifying pair can be constructed. Full proofs are given in (7). Detailed analysis of the algorithm implies that dragons have a so-called dragon-decomposition.

Definition 5.3 *A **dragon-decomposition** is given by a forest $F \subseteq E$ which has the following properties.*

1. *The components of forest $(V(F) \cup A, F)$ are exactly $\{F_a : \text{for each } a \in A\}$ s.t. for each $a \in A$ we have $A \cap V(F_a) = \{a\}$.*
2. *Let $\omega^F : V(F) \cup A \rightarrow \Omega$ be the (uniquely defined) function s.t. each edge in F is ω^F -balanced. Let F' be the set of ω^F -balanced edges. Let \mathcal{K} is the family of components of $G - F'$. F/\mathcal{K} is a tree.*
3. *$K, V(F) \cap V(K), \omega^F, \pi$ is critical.*

Lemma 5.4 *Dragons are exactly those p -graphs which have a dragon-decomposition. $V(F) \cup A$ is exactly the set of uniquely reachable nodes.*

References

- [1] M. Chudnovsky, J.F. Geelen, B. Gerards, L. Goddyn, M. Lohman, P. Seymour, *Packing non-zero A -paths in group-labeled graphs*, submitted
- [2] M. Chudnovsky, W.H. Cunningham, J.F. Geelen, *An algorithm for packing non-zero A -paths in group-labeled graphs*, manuscript
- [3] T. Gallai, *Maximum-minimum Sätze und verallgemeinerte Faktoren von Graphen*, Acta Mathematica Academiae Scientiarum Hungaricae 12 (1961) 131-137.
- [4] L. Lovász, *Matroid matching and some applications*, Journal of Combinatorial Theory Ser. B 28 (1980) 208-236.
- [5] W. Mader, *Über die Maximalzahl kreuzungsfreier H -Wege*, Archiv der Mathematik (Basel) 31 (1978) 387-402.
- [6] G. Pap, *Packing non-returning A -paths*, submitted
- [7] G. Pap, *Packings and 2-packings of non-returning A -paths in p -graphs*, manuscript
- [8] A. Schrijver, *A short proof of Mader's \mathcal{S} -paths theorem*, Journal of Combinatorial Theory Ser. B 85 (2001) 319-321.
- [9] A. Sebő, L. Szegő, *The path-packing structure of graphs*, in: Integer Programming and Combinatorial Optimization (Proceedings of 10th IPCO Conference, New York, 2004; D. Bienstock, G. Nemhauser, eds.) Lecture Notes in Computer Science 3064, Springer-Verlag Berlin Heidelberg (2004) 256-270.

