

部分順列グラフの集合を表現する ZDD の構築

高嶋 勇哉[†] 川原 純[‡] 湊 真一[§]
 京都大学[†] 京都大学[‡] 京都大学[§]

1 はじめに

部分グラフの列挙問題は、与えられたグラフに対して特定の部分グラフを列挙する問題である。この問題を解く手法として、ゼロサプレス型二分決定グラフ (Zero-suppressed Binary Decision Diagram; ZDD) と呼ばれるデータ構造を用いた手法がある [1]。本稿では与えられたグラフの部分順列グラフ集合を表現する ZDD を構築するアルゴリズムを新たに提案する。

2 準備

ZDD は場合分け二分木を圧縮した、集合族をコンパクトに表現可能なデータ構造である [1]。あるグラフ G の部分グラフ集合を ZDD で表現するため、 G の任意の辺誘導部分グラフは G の辺集合の部分集合として表現できることを利用し、ZDD の各節点には G の辺をラベルとして割り当てる。これによって G の辺集合に対する集合族、つまり G の部分グラフ集合を ZDD で表現できる。ZDD を用いることで解の個数に直接依存しない計算時間で処理を行うことができるため、膨大な解を持つ問題でも現実的な時間で列挙が完了することが多い。また ZDD 上では各種の集合演算を効率的に行えるため、得られた解集合に対する追加の制約による絞り込みなども効率的に行える。

ZDD を構築する汎用的で効率的な手法としてフロンティア法がある [2]。フロンティア法はトップダウンに ZDD を構築する手法であり、 s - t パス、サイクル、森など、多くの主要なグラフクラスについて、与えられたグラフの部分グラフ集合を表す ZDD を構築できる。フロンティア法による部分グラフの

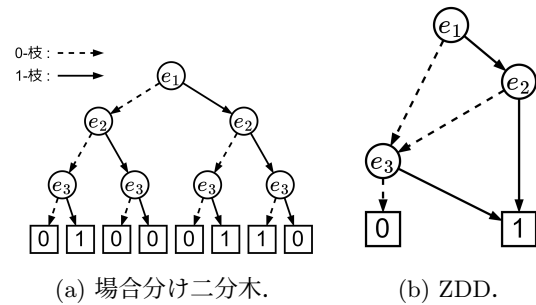


図 1: 場合分け二分木と ZDD.

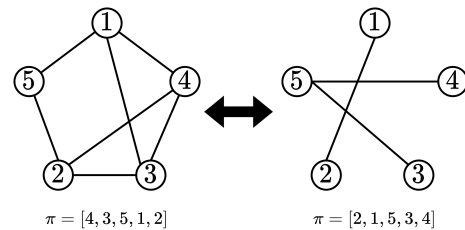


図 2: 順列グラフとその補グラフ.

列挙では、構築中の ZDD の各節点に、その節点が表示部分グラフにおける各頂点の度数に関する情報や連結成分に関する情報などを持たせ、それらの情報を利用した枝刈をしながら目的とする ZDD を幅優先的に構築する。

順列 $\pi = [\pi_1, \pi_2, \dots, \pi_n]$ に対して、 $V = \{1, 2, \dots, n\}$, $E = \{(i, j) \mid i, j \in V, (i - j)(\pi_i^{-1} - \pi_j^{-1}) < 0\}$ で定義される無向グラフ $G = (V, E)$ を π の順列グラフという。ここで π_i^{-1} は数字 i が順列 π においてどの位置に出現するかを表す。図 2 に順列グラフの例を示す。順列グラフでは、その補グラフが逆順の順列に対応する順列グラフになるという性質がある。

ある無向グラフ $G = (V, E)$ が順列グラフであるかどうかは、 G に対する permutation labeling が存在するかどうかと言い換えることができる。permutation labeling とは特別な頂点ラベリングである。全単射 $L: V \rightarrow \{1, 2, \dots, n\}$ に対して、 $(x, y) \in E \Leftrightarrow$

Construction of ZDD representing a set of permutation subgraphs

[†] Yuya Takashima, Kyoto University

[‡] Jun Kawahara, Kyoto University

[§] Shin-ichi Minato, Kyoto University

$[L(x) - L(y)][\pi^{-1}(L(x)) - \pi^{-1}(L(y))] < 0$ という条件を満たすような $\{1, 2, \dots, n\}$ の順列 π が存在するとき、 L を G の permutation labeling という。また permutation labeling に対して、以下の定理が成り立つことが知られている。

定理 1 ([3]). 写像 $F : x \mapsto L(x) - d^-(x) + d^+(x)$ ($x \in V$) が単射であるとき、またそのときに限って、全単射 $L : V \rightarrow \{1, 2, \dots, n\}$ は G の permutation labeling になる。ただし、 $\text{Adj}(x)$ を頂点 x に隣接する頂点の集合として、 $d^-(x) = |\{y \in \text{Adj}(x) \mid L(y) < L(x)\}|$ 、 $d^+(x) = |\{y \in \text{Adj}(x) \mid L(y) > L(x)\}|$ である。

3 提案手法

入力グラフ $G = (V, E)$, $E = \{e_1, \dots, e_m\}$ に対して、部分順列グラフ集合を表現する ZDD をフロンティア法で構築する手法について述べる。以降の議論では ℓ ($\leq |V|$) 頂点からなる部分順列グラフ集合を表現する ZDD を構築する手法について考える。 G の全ての部分順列グラフ集合は、上記の手法を頂点数 $\ell = 1, \dots, |V|$ に対して適用し、それらの和集合を取ることで得ることができる。フロンティア法では始めに G の辺 e_1 をラベルとして与えられた根節点を作成し、 e_1 を部分グラフの辺として含めるかどうかの場合分けを意味する 2 つの枝を伸ばして、続く子節点を作成する。深さ i の節点では辺 e_i がラベルとして割り当てられており、幅優先的に同様に子節点を作成していく。このとき各節点では根節点からのただ 1 つのパスが存在し、それがあある 1 つの辺誘導部分グラフを表現しているので、その部分グラフが ℓ 頂点の順列グラフとしての条件を満たしているか随時判定する。

まず、部分グラフが ℓ 頂点からなるという条件を課す方法を説明する。ラベル e_i の節点の 1-枝の先の子節点は e_i が部分グラフに使われていることを意味するので、 e_i の 2 つの端点の部分グラフとしての次数はそれぞれ 1 ずつ増える。次数が 1 以上の頂点がちょうど ℓ 個であれば、辺誘導部分グラフが ℓ 頂点をもつ（孤立点は無視する）。各節点には、節点に対応する部分グラフの各頂点が次数 1 以上か 0 かを記憶させる。次数 1 以上の頂点数が ℓ を超えた時点で、枝の先を 0-終端に接続する（枝刈と呼ぶ）。最終的に次数 1 以上の頂点数が ℓ 未満の場合も枝刈する。

次に、部分グラフに permutation labeling が存在するかの判定の概要を述べる。 ℓ 頂点部分グラフの permutation labeling の候補は $\ell!$ 個存在する。ZDD の各節点には、対応する部分グラフにおいて矛盾しない permutation labeling の候補を（最大 $\ell!$ 個）すべて格納する。各節点の更新時に矛盾する permutation labeling 候補は消去する。permutation labeling の候補が 0 個になった時点で、部分順列グラフにならないことが確定するので、枝刈する。

各節点には、具体的には $\{(P_j, c_j, F_j)\}_{j=1, \dots, \ell!}$ を格納する。 P_j は $1, \dots, \ell$ を並び替えた順列である。 $P_1, \dots, P_{\ell!}$ は $1, \dots, \ell$ のすべての順列である。この順列の意味は、 h 番目は、部分グラフの辺を添字の小さな順に並べた際に、初めて現れるのが h 番目である頂点のラベリングの値という意味である。すなわち P_j は permutation labeling の候補の 1 つとみなせる。 c_j は P_j が permutation labeling として矛盾していれば 0、矛盾していなければ 1 である。 F_j は定理 1 に現れる F であり、 F の単射性を保証するためのものである。詳細は省略する。

通常の前ティア法と同様、2 つの節点が等価、すなわち節点ラベルと格納されている情報がすべて等しいとき、2 つの節点を併合する。これにより重複した計算を回避でき、ZDD 構築の時間や記憶量が削減される。様々な詳細は省略しているが、以上の手法で所望の ZDD を構築できる。

4 結論

フロンティア法を用いて、部分順列グラフ集合を表現する ZDD を構築するアルゴリズムを新たに提案した。計算量の改善などが今後の課題である。

本研究の一部は、JSPS 科研費 JP20H00605, JP20H05794, JP20H05964 の助成を受けたものです。

参考文献

- [1] S. Minato. Zero-Suppressed BDDs for Set Manipulation in Combinatorial Problems. In *Proc. of DAC*, pages 272–277, 1993.
- [2] J. Kawahara et al. Frontier-Based Search for Enumerating All Constrained Subgraphs with Compressed Representation. *IEICE Trans.*, E100.A(9):1773–1784, 2017.
- [3] M.C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Elsevier Science, 2004.