数理解析研究所講究録
950巻 1996年 240-245

近似解をも考慮に入れた多項式時間変換

九州大学工学部　宮崎 修一　(Shuichi Miyazaki)
九州大学工学部　岩間 一雄　(Kazuo Iwama)

1　はじめに

組合せ問題間の多項式時間変換は，従来，NP 完全性の証明のように複雑さの構造を明らかにするために用いられてきた．しかし，最近では実際に問題を解くために利用する試みがなされている[2]．すなわち，問題 A を問題 B に多項式時間で変換して，問題 B に対するアルゴリズムを用いて解くことにより，問題 A を解くという手法である．このような方法が効果を発揮するためには，問題 B が，例えば CNF 論理式の充足可能性問題（SAT）のように，高速なアルゴリズムが研究されている問題でなければならない．また，多項式時間変換が，効率の良い（問題のサイズをあまり増大させない）ものである必要がある．我々は，3 つのグラフ問題から SAT への，変数の数を出来るだけ少なくするような変換を考案した [2]．

しかし，SAT の解法研究が進んでいるとは言え，完全な解を求めるのは以前として難しい．ところが，近年開発された局所探索法というアルゴリズムは，100 変数程度なら，充足可能な式に対して，充足不能となる節数が 3 ～ 5 程度の近似解を極めて高速に見つけるという実験結果が報告されている．そこで，我々の変換を近似解にも対応させることが出来れば，局所探索法を利用することにより，グラフ問題の近似解を高速に見つけることが出来る．

本稿では，k クリーク問題から SAT への多項式時間変換で，ある程度近似解にも対応できる変換を提案する．この変換は，グラフ G，整数 k から，CNF 論理式 f への変換で，以下の (i)，(ii) を満たす．(i) G が k クリークを持てば f は充足可能であり，持たなければ充足不能である．(ii) f 中の 0 になる節数が l であるような解が見つかれば，G 中に $\lceil \frac{2}{m(m+1)}(mk-l) \rceil$ 頂点からなるクリークが存在する．ただし m は $\frac{2l}{k} \leq m < \frac{2l}{k}+1$ を満たす整数である．

最適化問題間の変換としては，L-reduction[3]，PTAS-reduction[1] 等が知られている．これは，問題 A から問題 B に変換できたとすると，問題 B に対する，近似度が定数の多項式時間近似アルゴリズムが存在するならば，問題 A に対しても存在するという点で，近似度を保存する変換である．しかし，これらは非常に限定された問題どうしにしか適用できず，例えば，クリーク問題から SAT への変換は知られていない．

我々の変換は，この様な意味での近似度を保存するものではない．しかし，充足可能な式に対して，充足できない節の数が非常に小さく抑えられるような変数割り当てが得られるなら，それから得られるクリークのサイズも，k に近いものになる．

2 問題

本稿で取り扱う問題はSATとkクリーク問題の二つである．まず，判定の面から考えると，

充足可能性問題（SAT）：与えられたCNF論理式fを1にできる割当が存在するか？

k-クリーク問題：与えられたグラフ中にk個以上の頂点からなる完全部分グラフが存在するか？

となる．次に最適化問題の面から見ると，

MAX-SAT：与えられたCNF論理式f中の節を最大何個1にできるか？

MAX-クリーク問題：与えられたグラフ中に最大何個の頂点からなる完全部分グラフがあるか？

という問題になる．我々が考えようとしている変換は，(i) k-クリークが存在するグラフは充足可能な式に，存在しないグラフは充足不能な式に変換する，(ii) k-クリークが存在しないグラフから変換された式の解から，元の問題のクリークをなす頂点の個数が推測できる，というものである．

3 変換方法

グラフの頂点数をnとして，各頂点には0から$n-1$の番号が付けられているものとする．変換は変数$x_{i,j}$ $(1 \le i \le k, 1 \le j \le N, N = \lceil \log n \rceil)$，すなわち，$k\lceil \log n \rceil$個の変数を用いて行う．直観的には，$1 \le i \le k$という$k$個の箱を用意して，その箱にクリークとなる頂点を入れていくということを行う．変数割り当てが行われた時，箱iには$x_{i,N}, x_{i,N-1}, \cdots, x_{i,1}$に割り当てられる0，1の2進表現で表される頂点が入れられたことを意味する．例えば$N = 5$のときの例で考えると，$x_{i,5}, x_{i,4}, x_{i,3}, x_{i,2}, x_{i,1}$にそれぞれ$0,1,1,0,0$が割り当てられたということは，頂点12が選ばれたことに対応する．

変換に当たって以下のように定義を行う．変数への割り当てPに対して，$\boldsymbol{X}_i(P)$は$x_{i,N}, x_{i,N-1}, \cdots, x_{i,1}$に与えられた値の2進表現を表すものとする．例えば$N=5$のとき，割り当て$P$が$x_{4,5}, x_{4,4}, x_{4,3}, x_{4,2}, x_{4,1}$にそれぞれ$0,1,1,0,0$を割り当てるものであったとすると$\boldsymbol{X}_4(P) = 12$である．次に$C(\boldsymbol{X}_i = l)$は$\boldsymbol{X}_i(P) = l$となる全ての割り当て$P$に対して0となる節を表すものとする．例えば$N = 5$の場合$C(\boldsymbol{X}_3 = 9)$というのは，$(x_{3,5} + \overline{x_{3,4}} + x_{3,3} + x_{3,2} + \overline{x_{3,1}})$という節を表す．2つの節$C_1 = (y_1 + \cdots + y_i)$と$C_2 = (z_1 + \cdots + z_j)$に対し，$C_1 + C_2$は節$(y_1 + \cdots + y_i + z_1 + \cdots + z_j)$を表すものとする．

変換 kClique

ステップ1：$1 \le i < j \le k, 0 \le l \le n-1$の各$i,j,l$に対して，節$C(\boldsymbol{X}_i = l) + C(\boldsymbol{X}_j = l)$を作る．

ステップ2：$1 \le i < j \le k, 0 \le a \le N-1, 0 \le b \le N-1$の各$i,j,a,b$に対して，$a > n-1$または$b > n-1$ならば節$C(\boldsymbol{X}_i = a) + C(\boldsymbol{X}_j = b)$を作る．

ステップ3：$1 \le i < j \le k, 0 \le a < b \le n-1$の各$i,j,a,b$に対して，頂点$a$,$b$間に枝がなかったら2つの節$C(\boldsymbol{X}_i = a) + C(\boldsymbol{X}_j = b)$，$C(\boldsymbol{X}_i = b) + C(\boldsymbol{X}_j = a)$を作る．

前述したように，変数への真偽割り当てはグラフの中からk個の頂点を選ぶことに対応している．ただし，$\boldsymbol{X}_i(P) = \boldsymbol{X}_j(P)$となる割当や，$\boldsymbol{X}_i(P) > n-1$となる割り当ても存在するから，同じ頂点を重複して選んだり，実在しない頂点（番号がnから$N-1$）を選んでしまう可能性もある．つまり，変数への0，1の割り当ては，実在する頂点（以下実在頂点と呼ぶ）と実在しない頂点（以下仮

想頂点と呼ぶ）の中からちょうど k 個の頂点を，重複を許して選ぶことに対応している．

4 条件の考察

4.1 判定性の保存

まず，作られた節はどういうとき 0 となるかについて考える．ステップ 1 の節は，同じ実在頂点を重複して選んだときに 0 となる．また，ステップ 2 で作られる節は，仮想頂点を選んだときに 0 となる．最後に，ステップ 3 の節は間に枝のない 2 つの現実頂点を選んだとき 0 となる．

定理 1. 変換 kClique は，k クリークを持つ例題は充足可能な式に，持たない例題は充足不能な式に変換する．□

証明. k クリーク問題の例題 G, k が論理式 f に変換されるとする．まず，グラフ G が k クリークを持つと仮定する．その k クリークを構成する頂点を $cl_1, cl_2, \cdots, cl_k$ とする．これらはクリークであるから，当然どの二つも同じ番号ではないし，仮想頂点も含まれないし，どの 2 頂点間にも必ず枝が存在する．従って，$\boldsymbol{X}_1(P_1) = cl_1, \boldsymbol{X}_2(P_1) = cl_2, \cdots, \boldsymbol{X}_i(P_1) = cl_i, \cdots, \boldsymbol{X}_k(P_1) = cl_k$ となる割り当て P_1 はステップ 1 〜 3 のどの節をも 0 にしないため f の充足解である．よって，グラフ G が k クリークを持っているならば，論理式 f は充足可能である．

次に，作られた論理式 f が充足可能であると仮定する．そのときの充足解を P_2 とし，P_2 のもとで，$\boldsymbol{X}_1(P_2) = cl'_1, \boldsymbol{X}_2(P_2) = cl'_2, \cdots, \boldsymbol{X}_i(P_2) = cl'_i, \cdots, \boldsymbol{X}_k(P_2) = cl'_k$ であるとする．すると，P_2 が充足解であるという条件から，グラフ G の頂点集合 $cl'_1, cl'_2, \cdots, cl'_k$ には同じ頂点が 2 つ以上は含まれず，仮想頂点も存在せず，どの 2 頂点間にも枝が存在することになる．従ってグラフ G には $cl'_1, cl'_2, \cdots, cl'_k$ からなる k クリークが存在する．よって，論理式 f が充足可能ならば，グラフ G 中には k クリークが存在する．□

4.2 近似の性質の保存に関する考察

次に，k クリークがないグラフ G から充足不能な論理式 f が作られる場合に対して議論を行う．この時考えることは，SAT の解が与えられた時，そこから，元のグラフ中にあるクリークの頂点数を求めることである．まず，変数に適当に 0,1 を割り当てた時，作られた節のうちいくつが 0 になるかを考えてみる．ステップ 1 で作られる節は，実在頂点を重複して選んだ場合に 0 となる．このとき，例えば頂点 6 を箱 2，4，5 に入れるような割り当てだったとすると，ステップ 1 の節のうち，$C(\boldsymbol{X}_2 = 6) + C(\boldsymbol{X}_4 = 6)$，$C(\boldsymbol{X}_2 = 6) + C(\boldsymbol{X}_5 = 6)$，$C(\boldsymbol{X}_4 = 6) + C(\boldsymbol{X}_5 = 6)$ の 3 つの節が 0 になることになる．一般に，頂点 i を j 回重複して選んだら，0 となる節は ${}_jC_2$ 個になる．これを全ての $0 \leq i \leq n-1$ に関して足し合わせた個数がステップ 1 で作られた節のうち 0 となる数である．ステップ 3 では，実在頂点どうしで間に枝のないペアの数が 0 となる節数に一致する．ただし，頂点を重複して選んでいた場合には枝のない本数も増えることになる．例えば，頂点 2 と 5 の間には枝がなくて，頂点 2 を 3 回（例えば箱 1，2，5 に），頂点 5 を 2 回（例えば箱 3，7 に）選んだとすると，$C(\boldsymbol{X}_1 = 2) + C(\boldsymbol{X}_3 = 5)$，$C(\boldsymbol{X}_1 = 2) + C(\boldsymbol{X}_7 = 5)$，$C(\boldsymbol{X}_2 = 2) + C(\boldsymbol{X}_3 = 5)$，$C(\boldsymbol{X}_2 = 2) + C(\boldsymbol{X}_7 = 5)$，$C(\boldsymbol{X}_5 = 2) + C(\boldsymbol{X}_3 = 5)$，$C(\boldsymbol{X}_5 = 2) + C(\boldsymbol{X}_7 = 5)$ の 6 節が 0 になることになる．ステップ 2 では，仮想頂点が選ばれた時節は 0 となる．例えば，仮想頂点 15 が箱 10 に選ばれた場合を考えてみる．このとき，もし，残り $k-1$ 個は実在頂点ならば，$C(\boldsymbol{X}_1 =$

$cl_1) + C(\boldsymbol{X}_{10} = 15)$，$C(\boldsymbol{X}_2 = cl_2) + C(\boldsymbol{X}_{10} = 15)$，$\cdots$，$C(\boldsymbol{X}_9 = cl_9) + C(\boldsymbol{X}_{10} = 15)$，$C(\boldsymbol{X}_{11} = cl_{11}) + C(\boldsymbol{X}_{10} = 15)$，$\cdots$，$C(\boldsymbol{X}_k = cl_k) + C(\boldsymbol{X}_{10} = 15)$ という $k-1$ 節が 0 になる．更に，例えば箱 6 にも仮想頂点 17 が選ばれたとする．このとき，$C(\boldsymbol{X}_6 = 17) + \cdots$ という $k-1$ 節が 0 になるが，節 $C(\boldsymbol{X}_6 = 17) + C(\boldsymbol{X}_{10} = 15)$ は 2 度数えていることになる．

つまり，簡単に考えれば以下のようになる．与えられたグラフ（n 頂点，頂点番号は $0 \sim n{-}1$）に仮想頂点（頂点番号は $n \sim N-1$）を加え，これらは孤立頂点にする．その中から論理式への割り当て P に従って $\boldsymbol{X}_1(P), \boldsymbol{X}_2(P), \cdots, \boldsymbol{X}_i(P), \cdots, \boldsymbol{X}_k(P)$ という k 個の頂点を選ぶ．そして，$1 \leq i \leq k'$ に対し，$(\boldsymbol{X}_i(P), i)$ を頂点とするグラフ $G'(P)$ を考える．つまり，グラフ G の頂点番号に，選んだ箱の番号を添字としてつけるわけである．そして，グラフ G 中で頂点 a と頂点 b の間に枝があった場合に $G'(P)$ の頂点 (a,i) と頂点 (b,j) 間に枝をつける．(a,i) と (a,j) の間には枝はつけない．つまり，重複して選んだ頂点の間には枝はつけないものとする．当然，a が G の仮想頂点なら，(a,i) は $G'(P)$ の孤立頂点になる．このとき，$G'(P)$ で枝のない総数と，f で P のもとで 0 となる節数は一致する．

補題 1. k クリーク問題の例題 G, k を変換 kClique により変換した論理式を f とする．f の節のうち変数割り当て P で 0 になる節数はグラフ $G'(P)$ で枝のない本数と一致する． □

証明. P のもとで 0 となる節数を l，$G'(P)$ 中の枝のない数を e とする．まず P のもとで 0 となる節を考える．節は全て $C(\boldsymbol{X}_i = a) + C(\boldsymbol{X}_j = b)$ という形のものである．グラフ $G'(P)$ の構成方法より，$C(\boldsymbol{X}_i = a) + C(\boldsymbol{X}_j = b)$ が 0 になるということは，$\boldsymbol{X}_i(P) = a$，$\boldsymbol{X}_j(P) = b$ であるため，グラフ $G'(P)$ に頂点 (a,i) と (b,j) があることになる．この節がステップ 1 のものなら，$a = b$ であるから，(a,i) と (b,j) の間には枝はない．ステップ 2 のものであれば，a または b が仮想頂点なので，(a,i)，(b,j) 間に枝はない．ステップ 3 のものであれば，グラフ G では頂点 a，b 間に枝がないわけだから，(a,i) と (b,j) の間には枝はない．同じ節は 2 つ以上ないので，0 となる節 1 つに対して，$G'(P)$ 中に枝のない頂点ペアが 1 つ対応している．よって $l \leq e$ となる．

次に，$G'(P)$ で枝のない頂点ペアがあったとする．それを (a,i)，(b,j) とすると，これらの頂点が $G'(P)$ にあるということは $\boldsymbol{X}_i(P) = a$，$\boldsymbol{X}_j(P) = b$ である．もし，G で a または b が仮想頂点だった場合には，ステップ 2 で作られた $C(\boldsymbol{X}_i = a) + C(\boldsymbol{X}_j = b)$ が 0 になっている．次に a，b 共に G の実在頂点だった場合を考える．$a = b$ ならステップ 1 で $C(\boldsymbol{X}_i = a) + C(\boldsymbol{X}_j = b)$ という節が作られており，それが 0 になっている．$a \neq b$ だったら，(a,i)，(b,j) 間に枝がないということは，a，b 間に枝がないということになる．従って，ステップ 3 で節 $C(\boldsymbol{X}_i = a) + C(\boldsymbol{X}_j = b)$ が作られており，それが 0 になる．従って，枝のない頂点ペアには f の 0 となる節が 1 つ対応している．よって $e \leq l$ である．従って $e = l$ となる．

補題 2. k クリーク問題の例題 G, k を変換 kClique により変換した論理式を f とする．このとき論理式 f の任意の変数割り当て P に対し，$G'(P)$ に k' クリークが存在するなら，G に k' クリークが存在する． □

証明. $G'(P)$ に k' クリークがあるとする．それを $(a_1, i_1), (a_2, i_2), \cdots, (a_{k'}, i_{k'})$ とする．もし，a_p が G の仮想頂点なら (a_p, i_p) は $G'(P)$ の孤立頂点である．したがって，任意の p に対して a_p はグラフ G の実在頂点である．また，任意の p，q に対して，$a_p \neq a_q$ であり，G では a_p，a_q 間に枝がある．従って，グラフ G では $a_1, a_2, \cdots, a_{k'}$ がクリークになっている．よって，グラフ G は k' クリークを持つ．

補題 3. k 頂点の完全グラフから l 本の枝をどのように取り除いても，残ったグラフの中に存在するクリークのうち最大のものは，少なくとも $k' = \lceil \frac{2}{m(m+1)}(mk-l) \rceil$ 頂点を持っている．

ただし，m は，

$$\frac{2l}{k} \leq m < \frac{2l}{k} + 1, \tag{1}$$

を満たす整数である． □

証明. はじめに，k 頂点に $\frac{k(k-1)}{2} - l$ 本の枝を配置して，最大のクリークの頂点数が k' になるようにすることができることを示す．$k''' = \frac{2}{m(m+1)}(mk-l)$ と置く．まず，k 頂点を k' 個の頂点からなるグループ m 個と，$k - mk'$ 個の頂点からなるグループ 1 個に分割する．ただし m は（1）式を満たす整数とする．ここで，（1）式より

$$\frac{k}{m+1} \leq k''' < \frac{k}{m} \tag{2}$$

となる．$k' \geq k'''$ より，$k - mk' \leq k - mk''' \leq k''' \leq k'$ となり，$k - mk' \leq k'$ である．そして $\frac{k(k-1)}{2} - l$ 本の枝を次のように配置することを考える．まず，各グループを完全グラフにする．次に各グループの頂点に 1 番から k' 番までの番号を付け（1 つのグループは 1 番から $k - mk'$ 番まで），異なるグループ間で番号の違う頂点どうしを全て結ぶ．すると，$\frac{k(k-1)}{2}$ 本のうち枝のない部分は，同じ番号が付けられた異なるグループ間の頂点どうしなので，${}_{m+1}C_2(k-mk') +_m C_2(k'-(k+mk'))$=$mk - \frac{m(m+1)}{2}k' \leq mk - \frac{m(m+1)}{2}k''' = l$ となり，与えられた本数にするまでに何本か取り除かなければならない．取り除く本数は $l - (mk - \frac{m(m+1)}{2}k') = \frac{m(m+1)}{2}(k'-k''')$ であるが，$k' = \lceil k''' \rceil$ なので，$k' - k''' < 1$ であり，$\frac{m(m+1)}{2}(k'-k''') < \frac{m(m+1)}{2}$ となる．そこで，$k' \neq 1$ のときは，$m+1$ 個のグループ間には少なくとも 1 本ずつの枝があるので，少なくとも ${}_{(m+1)}C_2 = \frac{m(m+1)}{2}$ 本枝があるから，その中から必要なだけ取り除く．また，$k' = 1$ のときは，$k''' \leq 1$ であるが，$k''' < 1$ ならば（2）式より $\frac{k}{m+1} \leq k''' < 1$ となり，$m > k-1$ となるが，m は整数なので $m \geq k$．また，（1）式より，$\frac{k(m-1)}{2} < l$ であるが，$m \geq k$ から，$\frac{k(k-1)}{2} \leq \frac{k(m-1)}{2} < l$ となるが，枝の本数は $\frac{k(k-1)}{2} - l$ であるため，矛盾する．したがって $k' = 1$ のとき $k''' = 1$ となり，$k' - k''' = 0$ なのでこの場合は取り除く必要はない．ゆえに，k 頂点に枝を $\frac{k(k-1)}{2} - l$ 本配置して最大のクリークをなす頂点数が k' になるような配置は可能である．

つぎに，これよりも悪い配置があると仮定する．すなわち，k 頂点に $\frac{k(k-1)}{2} - l$ 本の枝を配置して最大のクリークの頂点数が $k''(< k')$ であるような配置があると仮定する．そのとき，その k 頂点のグラフで，クリークをなす頂点数が最大のものを k_1 個とする（$k_1 \leq k''$）．すると，残り $k - k_1$ 個の頂点のうち 1 つでも，その k_1 個の頂点全てに枝があれば $k_1 + 1$ クリークが存在したことになるので，k_1 個が最大のクリークであることに矛盾する．したがって，$k - k_1$ 個の各頂点は少なくとも 1 本は k_1 個の頂点に対して枝がない．よって少なくとも $k - k_1$ 本は枝がないことになる．

次に残りの $k - k_1$ 頂点のなかで最大のクリークの頂点数を k_2 とする（$k_2 \leq k'$）．やはり，残った $k - k_1 - k_2$ 頂点の中の 1 つの頂点でも k_2 頂点全てに枝があれば k_2 が最大であることに矛盾するので，$k - k_1 - k_2$ 本は枝がないことになる．

この操作を m 行なった後を考える．そのとき残っている頂点は $k - (k_1 + k_2 + \cdots + k_m)$ である．$k'' < k'$ であるが，k'' は整数なので $k'' \leq k' - 1$ である．また，$k' - k''' < 1$ なので，$k'' < k''' \leq k'$ となる．各 $k_i (1 \leq i \leq m)$ は $k_i \leq k''$ を満たすことより，$k - (k_1 + k_2 + \cdots + k_m) \geq k - mk'' >$

$k - mk''' > 0$ で，少なくとも m 回まではこの操作を続けられることが分かる．その段階で枝がない部分の総数は

$$\begin{aligned}
&(k-k_1)+(k-k_1-k_2)+\cdots+(k-k_1-k_2-\cdots-k_m)\\
&= mk-(mk_1+(m-1)k_2+\cdots+2k_{m-1}+k_m)\\
&\geq mk-\frac{m(m+1)}{2}k''\\
&> mk-\frac{m(m+1)}{2}k'''=l
\end{aligned}$$

となり，枝数が $\frac{k(k-1)}{2} - l$ 本あったことに矛盾する．したがって最大のクリークが k' よりも少ないような配置はない．ゆえに，どのように配置しても k' クリークは存在する． □

SAT で 0 となる節数が分かった場合，元のグラフ中のクリーク数が以下のように分かる．

定理 2. k クリーク問題の例題 G, k を変換 kClique により変換した論理式を f とする．このとき，f の 0 となる節数が l 個であることが分かれば，G 中の最大のクリークは，少なくとも $k' = \lceil \frac{2}{m(m+1)}(mk-l) \rceil$ 頂点を持っている．ただし，m は，$\frac{2l}{k} \leq m < \frac{2l}{k}+1$，を満たす整数である． □

証明. P のもとで論理式 f 中の 0 となる節数が l だったとする．補題 1 より，グラフ $G'(P)$ には枝が l 本ないことが分かる．補題 3 より，グラフ $G'(P)$ は k' クリークを持っている．補題 2 より，G は k' クリークをもっている． □

この変換の特徴は，論理式のどのような変数割り当てに対しても，作られた節のほとんどが充足されてしまうため，SAT の近似アルゴリズムに最適解からの差の精度が必要になってくる．$P \neq NP$ ならば，一般の SAT に対してはそのような近似アルゴリズムは存在しないが，我々の変換で得られた特殊な論理式に対する精度の良い近似アルゴリズムが存在すれば，我々の変換を用いてクリークの良い近似解を求められる可能性がある．

5 おわりに

本研究では，クリーク問題を判定問題，最適化問題の両面から見た場合の SAT への多項式時間変換について議論した．このとき，SAT の近似アルゴリズムの良さがクリークにあまり反映してこないという問題点が残った．今後は，この変換を，クリークを直接見つけるアルゴリズムに応用していくつもりである．

参考文献

[1] P. Crescenzi and A. Panconesi, "Completeness in approximation classes," *Information and Computation*, 93, pp. 241–262, 1991.

[2] K. Iwama and S. Miyazaki, "SAT-Variable complexity of hard combinatorial problems," *in* Proc.13th IFIP World Computer Congress, pp. 253–258, 1994.

[3] C. H. Papadimitriou, "Optimization, approximation, and complexity classes," *J. Comput. and System Sci.*, 43, pp. 425–440, 1991.