A new characterization for parity graphs and a coloring problem with costs*

Klaus Jansen
Max-Planck Institute for Computer Science
Im Stadtwald
66 123 Saarbrücken, Germany
jansen@mpi-sb.mpg.de

Abstract

In this paper, we give a characterization for parity graphs. A graph is a parity graph, if and only if for every pair of vertices all minimal chains joining them have the same parity. We prove that G is a parity graph, if and only if the cartesian product $G \times K_2$ is a perfect graph.

Furthermore, as a consequence we get a result for the polyhedron corresponding to an integer linear program formulation of a coloring problem with costs. For the case that the costs $k_{v,3} = k_{v,c}$ for each color $c \geq 3$ and vertex $v \in V$, we show that the polyhedron contains only integral 0/1 extrema if and only if the graph G is a parity graph.

1 Introduction

A graph is a *parity graph*, if and only if for every pair of vertices all minimal chains joining them have the same parity. Parity graphs are perfect [15] and are a subclass of the Meyniel graphs [11]. In a parity graph, each odd cycle of length at least five has two crossing chords. The class of parity graphs includes bipartite graphs and cographs. A polynomial algorithm for the recognition of parity graphs (based on three operations) is given in [3]. Furthermore, the problems maximum independent set, maximum clique, minimum coloring and minimum partition into cliques can be solved in polynomial time for these graphs (see also [3]). Recently, Cicerone and Di Stefano have given improved algorithms for the recognition, maximum weighted independent set and clique problem [5].

In this paper, we prove that G is a parity graph, if and only if the cartesian product $G \times K_2$ is a perfect graph. A partial characterization of the cartesian product $G \times K_2$ has been given already by Ravindra and Parathasarathy [13]. Independently, the characterization of parity graphs was found also by de Werra and Hertz [6]. We give a direct proof that $G \times K_2$ is a perfect graph for each parity graph G. Our proof contains also an interesting algorithm to color each induced subgraph of $G \times K_2$ using the combinatorial structure of the parity graphs. An indirect shorter proof (without giving an algorithm) was found by Reed [14].

*This work was done while the author was associated with the University Trier and supported in part by DIMACS and by EU ESPRIT LTR Project No. 20244 (ALCOM-IT).

Furthermore, we study an integer program formulation of a special case of the general optimum cost chromatic partition (GOCCP) problem. The GOCCP problem can be described as follows: An instance is given by an undirected graph $G = (V, E)$ with n vertices and by a $(n \times m)$ - cost-matrix $(k_{v,c})$ with unrelated costs $k_{v,c}$ to execute job v on machine c. The problem is to find a partition of the graph G into independent sets $U_1, \ldots, U_m$ such that $\sum_{c=1}^{m} \sum_{v \in U_c} k_{v,c}$ is minimum. A subproblem with only machine dependent costs $k_c = k_{v,c}$ for each $v \in V$ and $m = n$, called the OCCP problem, has been studied in [16, 10, 9]. In this paper, we consider the restricted case with $m = n$ and costs $k_{v,c} = k_{v,3}$ for $c \geq 3$ and $v \in V$. We prove that the polyhedron corresponding to the ILP contains only integral 0/1 extrema if and only if the graph G in the instance is a parity graph.

The OCCP problem restricted to circle and permutation graphs, introduced by Supowit [17], corresponds to a VLSI layout problem (see also [16]). Another application is given by Kroon et al. [10]. The OCCP problem for interval graphs is equivalent to the Fixed Interval Scheduling Problem (FISP) with machine dependent processing costs. In this scheduling problem each job $j \in J$ must be executed during a given time interval (s_j, f_j). We assume that a sufficient number of machines is available and that each job must be executed by one of the machines. If job j is executed by machine c, then the associated processing costs are k_c. The objective is to find a feasible schedule for all jobs with minimum total processing costs.

It is not difficult to see that the OCCP problem is NP-hard for arbitrary graphs. Sen et al. [16] proved that the OCCP problem for circle graphs is NP-hard. Moreover, they considered an integer linear program formulation of the OCCP problem. The polytope corresponding to the constraints of this ILP contains only integral (0/1) vertices, if the cartesian product $G \times K_n$ is a perfect graph. Sen et al. [16] proved that $G \times K_n$ is perfect if G is a tree and that there exists a perfect graph G such that $G \times K_3$ is not perfect.

Kroon et al. [10] studied the OCCP problem for interval graphs and trees. They showed that the problem restricted to trees can be solved in linear time and that the problem restricted to interval graphs is NP-hard even if there are only four different values for the coloring costs. For interval graphs G, they proved that the zero-one matrix corresponding to the constraints of the ILP is perfect, if and only if $G \times K_n$ does not contain an odd cycle of size 7 or more as induced subgraph.

It is proved that the cartesian product $G \times K_m$ for $m \geq 3$ is perfect, if and only if G is a diamond - free chordal graph (a generalization of forests) and that the GOCCP problem is solvable in polynomial time for these graphs [13, 9]. This implies that the polyhedron for the GOCCP problem with $m = n$ contains only integral extrema, if and only if the graph G in the instance is a diamond free chordal graph. Furthermore, we showed that the OCCP problem is NP-hard for bipartite graphs with four different cost values $(k_1 < k_2 < k_3 < k_4 = \ldots = k_n)$ and for permutation graphs with three different cost values. On the other hand, the OCCP problem is in P for cographs and the GOCCP problem is in P for graphs with constant treewidth.

2 Perfect matrices

In this section, we describe an integer linear program formulation of the restricted coloring problem with costs. Let I be an instance of the GOCCP problem containing a graph $G = (V, E)$

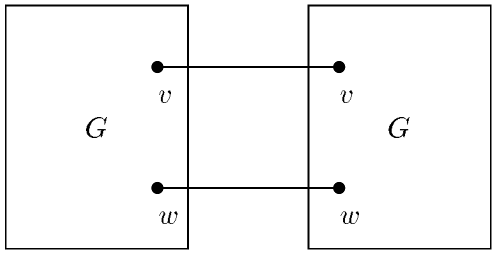


Figure 1: The cartesian product $G \times K_2$.

with n vertices and a $(n \times m)$ cost matrix $(k_{v,c})$ with $m = n$ and $k_{v,c} = k_{v,3}$ for $c \geq 3$ and $v \in V$. The objective function and the constraints of the problem can be described as follows:

$min \sum_{v \in V} [k_{v,1}\, x_{v,1} + k_{v,2}\, x_{v,2} + k_{v,3}(1 - x_{v,1} - x_{v,2})]$		(0)
$x_{v,1} + x_{v,2} \leq 1$	for each $v \in V$	(1)
$\sum_{v \in C} x_{v,c} \leq 1$	for each clique C in G, $1 \leq c \leq 2$	(2)
$x_{v,c} \in \{0, 1\}$	for each $v \in V, 1 \leq c \leq 2$	(3)

This coloring problem amounts to the decision problem which vertices of the graph receive colors 1 and 2. Once this is settled, the other vertices may receive colors $3, 4, \ldots, n$. For each vertex $v \in V$ and color $i \in \{1, 2\}$, the variable $x_{v,i}$ is equal 1, if vertex v receives color i. The total coloring costs are minimized by the objective function (0). The constraints (1) specify that each vertex v receives at most one of the colors 1, 2, and the constraints (2) guarantee that vertices that are connected by an edge are colored differently. Notice that the objective function is equivalent to the linear function

$$min \sum_{v \in V} [(k_{v,1} - k_{v,3})\, x_{v,1} + (k_{v,2} - k_{v,3})\, x_{v,2}].$$

The coefficient matrix (a zero-one matrix) corresponding to the restrictions $(1) - (2)$ is called M. A zero - one matrix M is called *perfect* if the polyhedron $P(M) = \{x | Mx \leq 1, x \geq 0\}$ has only integral extreme points. It follows that the GOCCP problem can be solved by applying a linear program algorithm, if the matrix M is perfect. The goal of this paper is a characterization of the graphs such that the polyhedron $P(M)$ contains only integral extrema.

The *cartesian product* $G_1 \times G_2 = (V_1 \times V_2, E)$ of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is defined by the edgeset

$$E = \{\{(u_1, u_2), (v_1, v_2)\} | [u_1 = v_1 \wedge \{u_2, v_2\} \in E_2] \vee [u_2 = v_2 \wedge \{u_1, v_1\} \in E_1]\}.$$

A graph $G = (V, E)$ is *perfect*, if and only if for each subset $V' \subset V$ the chromatic number $\chi(G[V'])$ of the subgraph $G[V']$ induced by V' is equal to the cardinality $\omega(G[V'])$ of a maximum clique in $G[V']$.

The cartesian product $G \times K_2$ of the original graph G and a complete graph K_2 with 2 vertices is illustrated in Figure 1. It is easy to see that the constraints (1), (2) for G defining

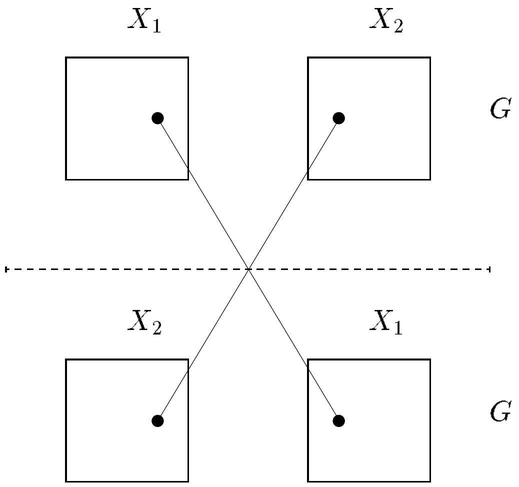


Figure 2: If $G = (X_1 \cup X_2, A)$ is bipartite, then $G \times K_2$ is bipartite.

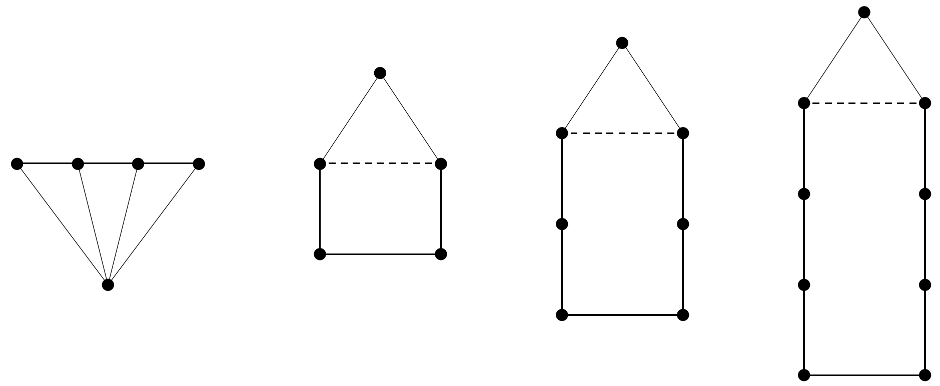
the matrix M can be seen as the clique inequalities for $G \times K_2$. The constraints (1) are for the cliques of size 2 consisting of pairs of corresponding vertices $(v,1),(v,2)$ of the two copies of G.

Furthermore, the constraints (2) correspond to the cliques in each of the copies of G. Thus, the polyhedron $P(M)$ has only integral extreme points (or equivalent M is perfect), if and only if $G \times K_2$ is perfect (Chvátal [4]). Therefore, the goal is to find a characterization of graphs G such that $G \times K_2$ is perfect. We note that $G \times K_2$ is a bipartite graph, if G is bipartite; see also Figure 2.

The solution x found by the linear program may not correspond directly to a solution of the coloring problem. If $k_{v,3} << min(k_{v,1}, k_{v,2})$ for all $v \in V$, then the best solution of the linear program is $x_{v,1} = x_{v,2} = 0$ for all $v \in V$. This implies that only machines $3, \ldots, n$ should be used in the best solution. If $\chi(G) \leq n-2$, then the solution found by the linear program represents also a solution of the coloring problem. If $\chi(G) > n-2$, then we have to color at least one vertex or two vertices with the colors $1, 2$. If $\chi(G) = n$ then G is a complete graph with n vertices, and the optimum solution of the coloring problem can be computed using a minimum weighted matching in a bipartite graph. If $\chi(G) = n-1$ then the complement of G does not contain two non-incident edges or a triangle. Using this assertion (and weighted matchings), an optimum solution can be found in polynomial time for $\chi(G) > n-2$.

3 Parity graphs

Let $\Gamma(x)$ be the set of vertices adjacent to x (the neighbours of x). Two vertices x and y are called *true twins*, if x and y are adjacent and have the same neighbours (that means $\Gamma(x) \cup \{x\} = \Gamma(y) \cup \{y\}$). Two vertices x and y that are not adjacent but have the same neighbours are called *false twins*. We define the *extension* of a graph G *by a bipartite graph* $B = (X_1 \cup X_2, A)$ the operation that generates a new graph by identification of a subset X of vertices of X_1 with a set of false twins of G (possibly with $|X| = 1$).



C_5^{**} $\quad$ C_5, C_5^* $\quad$ C_7, C_7^* $\quad$ C_9, C_9^*

Figure 3: A partial list of forbidden subgraphs.

Theorem 3.1 *[3] Every connected parity graph $G = (V, E)$ is obtained from a single vertex by the following operations:*

(1) ϕ_1 creation of a false twin,

(2) ϕ_2 creation of a true twin,

(3) ϕ_3 extension by a bipartite graph.

Parity graphs are also described by a list of forbidden induced subgraphs [3]. The list consists of the odd cycles C_{2k+1} $(k \geq 2)$ or odd cycles having a short chord C^*_{2k+1} (called generalized house) and the cycle C_5 with two non-crossing chords C_5^{**} (called gem). An interrupted line in Figure 3 indicates that the edge may be either present (C^*_{2k+1}) or not (C_{2k+1}). A metric characterization of parity graphs is given in [2] and parallel algorithms for recognition of parity graphs are presented in [1, 12].

4 Main theorem

In this section, we prove the following result:

Theorem 4.1 *The following two statements are equivalent:*

(1) *$G \times K_2$ is a perfect graph.*

(2) *G is a parity graph.*

Using this characterization and the theorem of Chvátal [4] (see also the section about perfect matrices), we obtain directly:

Corollary 4.1 *Let I be an instance of the GOCCP problem containing a graph $G = (V, E)$ with n vertices and coloring costs $k_{v,c}$ such that $k_{v,c} = k_{v,3}$ for $c \geq 3$ and $v \in V$. Then, we have the following equivalence: The polyhedron $P(M)$ corresponding to the instance I contains only integral extrema, if and only if G is a parity graph.*

The same result holds also for the OCCP problem with a sequence of coloring costs $k_1 < k_2 < k_3 = \ldots = k_n$ (in this case we have only a simpler objective function). Furthermore, we note that the number of cliques is exponentional in the number of vertices even in a cograph. Since $G \times K_2$ is a perfect graph, the strong optimization problem to find a vector that minimizes a linear function on $P(M)$ is solvable in polynomial time [8]. This implies the following result:

Corollary 4.2 *The GOCCP problem restricted to parity graphs and coloring costs $k_{v,c}$ such that $k_{v,c} = k_{v,3}$ for $c \geq 3$ and $v \in V$ can be solved in polynomial time.*

4.1 First Direction

In this subsection, we prove the first part of the main theorem:

Theorem 4.2 *If G is not a parity graph, then the cartesian product $G \times K_2$ is not perfect.*

Proof: To prove this, we have to consider the forbidden subgraphs. If G is no parity graph, then G must contain one of the following induced subgraphs:

- an odd cycle C_{2k+1} with $k \geq 2$,
- a generalized house C^*_{2k+1} with $k \geq 2$,
- a gem C^{**}_5.

In all three cases, we obtain an odd cycle in the cartesian product $G \times K_2$ and, therefore, we get a non-perfect induced subgraph in $G \times K_2$. This shows that $G \times K_2$ is not perfect if G is not a parity graph.

Case 1: If we have an odd cycle in G, then we have already a non-perfect induced subgraph in G and, therefore, also in $G \times K_2$.

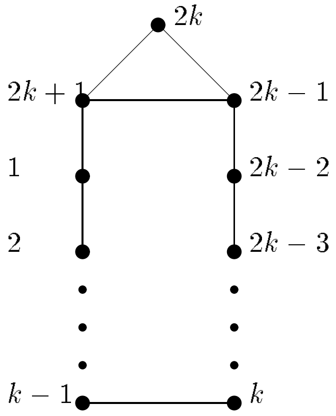
Case 2: A generalized house C^*_{2k+1} generates an odd cycle C_{2k+3} in $G \times K_2$; see also Figure 4.

Case 3: A gem C^{**}_5 generates an odd cycle C_7 in $G \times K_2$; see also Figure 5. □

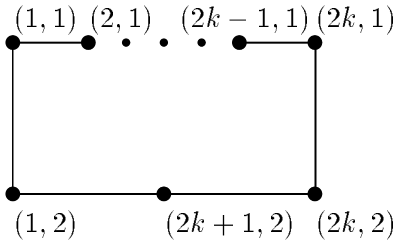
4.2 Second Direction

In this subsection, we prove the second part of the main theorem. An example for the algorithm to compute the colorings is given in the next section.

Theorem 4.3 *If G is parity graph, then the cartesian product $G \times K_2$ is perfect.*

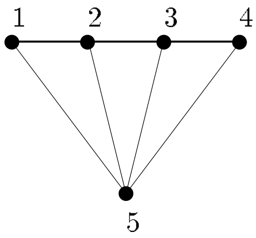


house C^*_{2k+1}

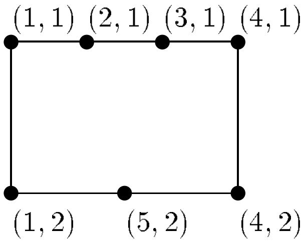


cycle C_{2k+3}

Figure 4: A house C^*_{2k+1} generates an odd cycle C_{2k+3} in $G \times K_2$.



gem C^{**}_5



cycle C_7

Figure 5: A gem C^{**}_5 generates a C_7 in $G \times K_2$.

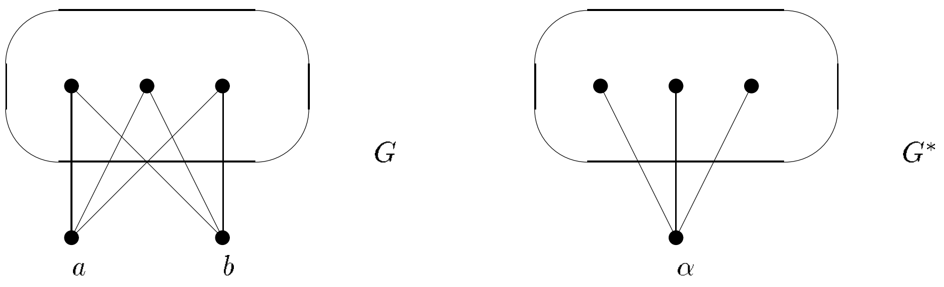


Figure 6: The transformation for false twins a and b.

Proof: Let $H = (V_H, E_H)$ be an induced subgraph of $G \times K_2$ with maximum clique size $\omega(H) = k$. Furthermore, let V_i be the set of vertices $\{v \in V | (v, i) \in V_H\}$ in the i.th part of $G \times K_2$, for $1 \leq i \leq 2$. If $k = 1$ then H contains only isolated vertices and can be colored with one color. Therefore, we may assume that $k > 1$. Moreover, we may assume that G is a connected parity graph; otherwise we compute a coloring for each corresponding component of $G \times K_2$.

In the following, we construct a k-coloring for the induced subgraph H. Simultaneously, we compute two k-colorings f_i for the induced subgraphs $G[V_i]$ $(1 \leq i \leq 2)$ such that $f_1(v) \neq f_2(v)$ for each vertex $v \in V_1 \cap V_2$. This gives a k-coloring for the graph H and proves the theorem.

Clearly, for $k > 2$, a maximum clique C lies either in $G[V_1]$ or in $G[V_2]$. We compute a k-coloring using weight functions $c^{(1)}, c^{(2)} : V \to \mathbb{N}_0$. At the beginning, we define

$$c^{(i)}(v) = \begin{cases} 1 & \text{if } v \in V_i \\ 0 & \text{otherwise.} \end{cases}$$

Then, the maximum weighted clique in G with weights $c^{(i)}(v)$ is equal to the maximum clique size $\omega(G[V_i])$, for $1 \leq i \leq 2$.

By the reverse operations ϕ_1^{-1}, ϕ_2^{-1} (and also ϕ_3^{-1}), we can transform the parity graph into a smaller parity graph. Using these reverse operations, we modify the weights of the vertices. In general, a weight $c^{(i)}(\alpha)$ stores the size of a maximum clique for a graph corresponding to α.

Operation ϕ_1^{-1}: False twins a and b in G. In this case, we transform G into a graph G^* (see also Figure 6) with weights $c^{(i)}(\alpha) = max(c^{(i)}(a), c^{(i)}(b))$.

Operation ϕ_2^{-1}: True twins a and b in G. In this case, we transform G into a graph G^* (see also Figure 7) with weights $c^{(i)}(\alpha) = c^{(i)}(a) + c^{(i)}(b)$.

Operation ϕ_3^{-1}: Extension by a bipartite graph $B = (X_1 \cup X_2, A)$ (see Figure 8). In this case, we remove the bipartite graph B and get a vertex α combining the false twins a and b with weights $c^{(i)}(\alpha) = max[c^{(i)}(a), c^{(i)}(b)]$. Notice that we store in vertex α only the weight of the vertices a and b (the number of colors for a and b). The cardinality of the maximum clique in G (or the minimum number of colors to color G) is given by the maximum of

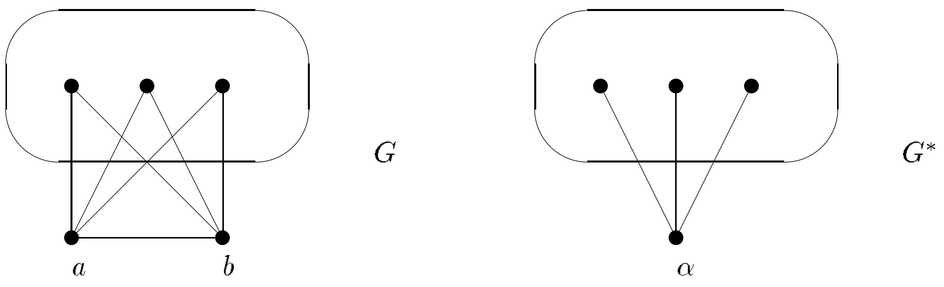


Figure 7: The transformation for true twins a and b.

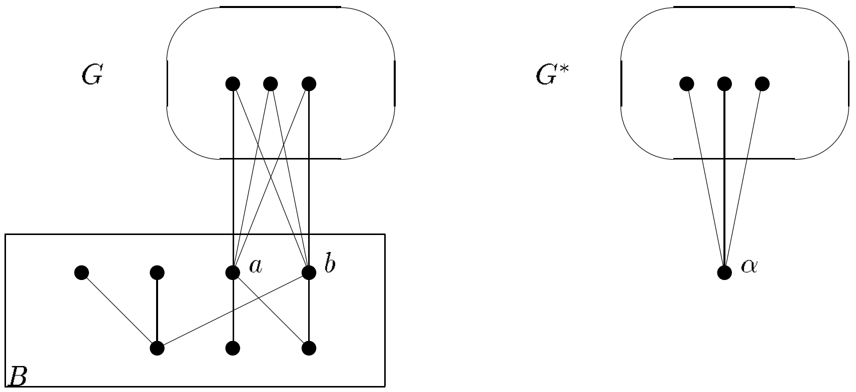


Figure 8: The transformation for an extension by a bipartite graph B.

(1) the weight of a maximum weighted clique in B,

(2) the weight of a maximum weighted clique in G^*.

In general, we replace the set $X \subset X_1$ of false twins by a vertex α with weights $c^{(i)}(\alpha) = max_{x \in X} c^{(i)}(x)$.

Recursively, we compute for parity graphs $G' = (V', E')$ with weights $c^{(i)}(v)$ for $v \in V'$ colorings $g_i : V' \to 2^{\{1,\ldots,k\}}$ such that the following invariants are satisfied:

(1) the cardinalities $|g_i(v)| = c^{(i)}(v)$,

(2) if $c^{(1)}(v) = c^{(2)}(v) = 1$ then $g_1(v) \neq g_2(v)$,

(3) if $c^{(1)}(v) = c^{(2)}(v) = k - 1$ then $g_1(v) \neq g_2(v)$.

This means that we compute for each vertex α a color set $g_i(\alpha)$ with cardinality equal to the weight $c^{(i)}(\alpha)$. Since the weight is equal to the size of a maximum clique corresponding to α, this color set stores a coloring of the graph corresponding to α. For the original graph $G = (V, E)$ the weights $c^{(i)}(v)$ are zero or one. Using invariant (2) we obtain color sets $g_i(v)$ with cardinalities one for the vertices v in $G[V_i]$ such that $g_1(v) \neq g_2(v)$ for $v \in V_1 \cap V_2$. The invariant (3) is used to split color sets for operation ϕ_3 (extension by a biparite graph).

Suppose that vertex α has the weight $c^{(i)}(\alpha) = k - 1$ in both graphs $G[V_i]$, $1 \leq i \leq 2$. Furthermore, suppose that the color sets are equal: $g_1(\alpha) = g_2(\alpha)$. If $X = \{a\}$ and vertex a is adjacent to a vertex $x \in X_2$ (in the bipartite graph B) with weights $c^{(i)}(x) = 1$, then we must color x with the same color. To avoid such a situation, we use the invariant (3).

In what follows, we compute recursively colorings g_i such that the invariants are satisfied. It must be noted that the sets $g_i(v)$ for the vertices v in $G[V_i]$ form a feasible coloring. We start with a single vertex graph and use color sets $g_i(v)$ of size equal to the weights $c^{(i)}(v)$. For true twins a, b, we distribute the color sets $g_i(\alpha)$ to both twins. Since we do not use one color for both a and b, this generates a feasible coloring. For operation ϕ_3, we extend the colorings of G^* to the vertices in B such that adjacent vertices x, y in B get color sets with $g_i(x) \cap g_i(y) = \emptyset$.

For a single vertex graph, we can find colorings g_i such that the invariants are satisfied for each $k \geq 2$. We assume now that we have such colorings g_i for the parity graph G^*, and apply one of the operations ϕ_1, ϕ_2 or ϕ_3 back to get G.

Operation ϕ_1: False twins a and b. This is a special case of ϕ_3 with $X_2 = \emptyset$ and $X_1 = X = \{a, b\}$.

Operation ϕ_2: True twins a and b. In this case, we have to split the color sets $g_1(\alpha)$ and $g_2(\alpha)$ such that the invariants are satisfied for the vertices a and b.

Case 1: We have to use two color sets of size $k - 1$ at vertex a or b. We may assume that $c^{(1)}(a) = c^{(2)}(a) = k - 1$.

Case 1.1: $c^{(1)}(b) = c^{(2)}(b) = 1$. In this case, $c^{(1)}(\alpha) = c^{(2)}(\alpha) = k$. We choose two different colors $red, blue \in \{1, \ldots, k\}$ and define the color sets as follows:

$g_1(a)$	$g_1(b)$	$g_2(a)$	$g_2(b)$
$\{1, \ldots, k\} \setminus \{red\}$	$\{red\}$	$\{1, \ldots, k\} \setminus \{blue\}$	$\{blue\}$

These color sets satisfy the invariants.

Case 1.2: $c^{(1)}(b) = 1$ and $c^{(2)}(b) = 0$. In this case, we have $|g_2(\alpha)| = k - 1$ and know the unique color $red \in \{1, \ldots, k\} \setminus g_2(\alpha)$. Using a color $blue$ different from red, we define the following color sets:

$g_1(a)$	$g_1(b)$	$g_2(a)$	$g_2(b)$
$\{1, \ldots, k\} \setminus \{blue\}$	$\{blue\}$	$g_2(\alpha)$	$\emptyset$

Again, these color sets satisfy the invariants.

Case 1.3: $[c^{(1)}(b) = 0$ and $c^{(2)}(b) = 1]$ or $[c^{(1)}(b) = c^{(2)}(b) = 0]$. The first case is symmetrical to case 1.2 and in the second case, we can use the color sets $g_i(a) = g_i(\alpha)$.

Case 2: We have to use two color sets of size 1 at vertex a or b. We may assume that $c^{(1)}(a) = c^{(2)}(a) = 1$.

Case 2.1: $c^{(1)}(b) = c^{(2)}(b) = 1$. If $g_1(\alpha) \cap g_2(\alpha) = \emptyset$, then we can split the color sets $g_1(\alpha)$ and $g_2(\alpha)$ arbitrarily. Suppose that $red \in g_1(\alpha) \cap g_2(\alpha)$. In this case, we define

$g_1(a)$	$g_1(b)$	$g_2(a)$	$g_2(b)$
$\{red\}$	$g_1(\alpha) \setminus \{red\}$	$g_2(\alpha) \setminus \{red\}$	$\{red\}$

and obtain colorings that satisfy our invariants.
Case 2.2: $c^{(1)}(b) = c^{(2)}(b) = k-1$. This case is symmetrical to case 1.1.
Case 2.3: $c^{(1)}(b) = c^{(2)}(b) = 0$. Here, we use the same color sets $g_i(a) = g_i(\alpha)$.
Case 2.4: $c^{(1)}(b) = 0$ and $c^{(2)}(b) \neq 0$. Here, we choose a color $red \in g_2(\alpha) \setminus g_1(\alpha)$ and define

$g_1(a)$	$g_1(b)$	$g_2(a)$	$g_2(b)$
$g_1(\alpha)$	$\emptyset$	$\{red\}$	$g_2(\alpha) \setminus \{red\}$

Case 2.5: $c^{(1)}(b) \geq 1$ and $c^{(2)}(b) > 1$ and not $c^{(1)}(b) = c^{(2)}(b) = k-1$. In this case, we have only a possible conflict at vertex a. Here, we choose two different colors $red \in g_1(\alpha)$ and $blue \in g_2(\alpha)$ and obtain feasible color sets using

$g_1(a)$	$g_1(b)$	$g_2(a)$	$g_2(b)$
$\{red\}$	$g_1(\alpha) \setminus \{red\}$	$\{blue\}$	$g_2(\alpha) \setminus \{blue\}$

Clearly, the case $c^{(1)}(b) > 1$ and $c^{(2)}(b) \geq 1$ works as well.

Case 3: It remains the case $[c^{(1)}(a) \neq c^{(2)}(a)$ OR $c^{(1)}(a) = c^{(2)}(a) \notin \{1, k-1\}]$ AND $[c^{(1)}(b) \neq c^{(2)}(b)$ OR $c^{(1)}(b) = c^{(2)}(b) \notin \{1, k-1\}]$. In this case, we can choose an arbitrary splitting of the color sets; e.g. take the first $c^{(i)}(a)$ colors of $g_i(\alpha)$ for $g_i(a)$ and the remaining colors for $g_i(b)$.

In all these cases we have obtained color sets for a and b that satisfy our invariants $(1)-(3)$.

Operation ϕ_3: Extension by a bipartite graph $B = (X_1 \cup X_2, A)$ where $X \subset X_1$ is a set of false twins. Since the maximum clique in $G[V_i]$ is at most k, we can use the fact that the weights $c^{(i)}(\alpha) \leq k$, $c^{(i)}(x) \leq k$ for each $x \in X_1 \cup X_2$ and that the sums $c^{(i)}(x) + c^{(i)}(y) \leq k$ for each edge $\{x, y\} \in A$.

Again, we consider by case analysis the color sets $g_1(\alpha)$ and $g_2(\alpha)$ and extend the coloring of G^* to the bipartite graph B. In all cases, (using an ordered list of colors) we color first the vertices in X_1 as early as possible and, then, the vertices in X_2 as late as possible. This implies that a vertex $x \in X_1$ with $c^{(i)}(x) = 1$ gets the first color and that a vertex $x \in X_2$ with $c^{(i)}(x) = 1$ gets the last color. Each vertex $x \in X_1 \cup X_2$ gets exactly $c^{(i)}(x)$ colors. We note that we first must take the colors in $g_i(\alpha)$; otherwise we can generate a conflict in G^*.

Case 1: There is a color $red \in g_1(\alpha) \setminus g_2(\alpha)$. For $G_1 = G[V_1]$, we start with color red and, then, take the colors in $g_1(\alpha) \setminus \{red\}$ and, finally, the colors in $\{1, \ldots, k\} \setminus g_1(\alpha)$. For $G_2 = G[V_2]$, we color the vertices with colors in the order $g_2(\alpha)$, $\{1, \ldots, k\} \setminus [\{red\} \cup g_2(\alpha)]$, $\{red\}$.

In this case, a vertex $x \in X_1$ with weights $c^{(1)}(x) = c^{(2)}(x) = 1$ gets the color sets $g_1(x) = \{red\}$ and $g_2(x) \neq \{red\}$. Furthermore, a vertex $x \in X_2$ with $c^{(1)}(x) = c^{(2)}(x) = 1$ has the color sets $g_2(x) = \{red\}$ and $g_1(x) \neq \{red\}$. For a vertex $x \in X_1$ with $c^{(1)}(x) = c^{(2)}(x) = k-1$, we have $red \in g_1(x) \setminus g_2(x)$ (since red is the last color for $G[V_2]$). The opposite statement holds for a vertex $x \in X_2$ with weights $c^{(1)}(x) = c^{(2)}(x) = k-1$.

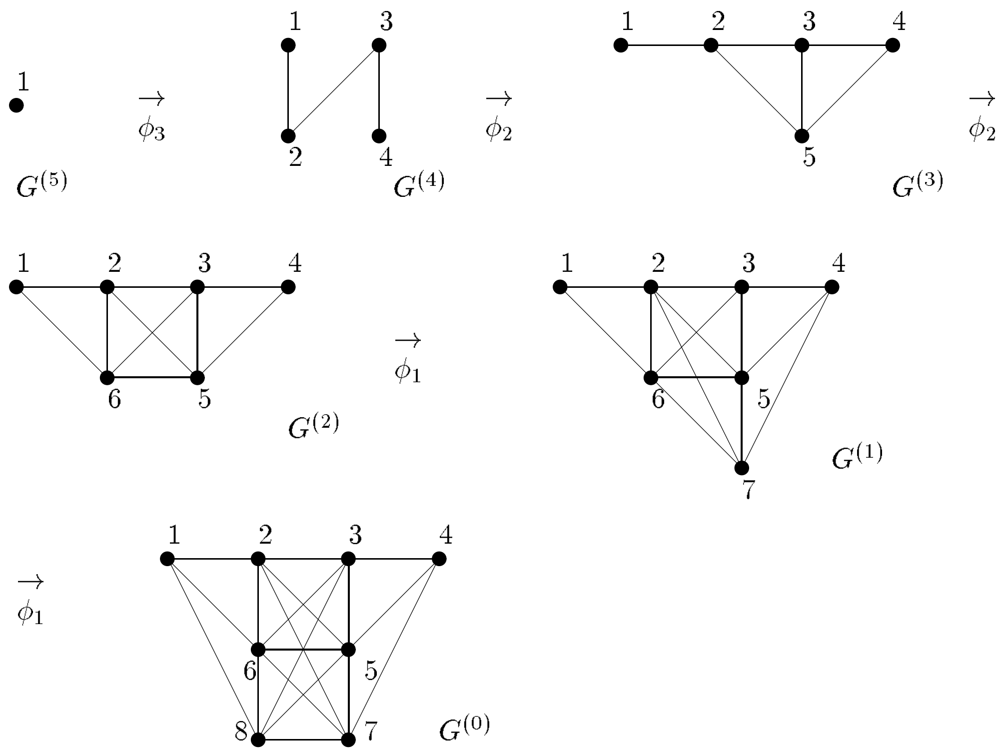


Figure 9: The generation of a parity graph $G = G^{(0)}$ using the operations ϕ_1, ϕ_2 and ϕ_3.

The case with a color $red \in g_2(\alpha) \setminus g_1(\alpha)$ works as well.

Case 2: The color sets $g_1(\alpha) = g_2(\alpha)$ and $2 \le c^{(1)}(\alpha) = c^{(2)}(\alpha) \le k-2$. In this case, there are at least two different colors $red, blue \in g_1(\alpha) = g_2(\alpha)$. Furthermore, we have two other colors $orange, yellow \in \{1, \ldots, k\} \setminus g_1(\alpha) = \{1, \ldots, k\} \setminus g_2(\alpha)$. For $G_1 = G[V_1]$, we use the following order of the colors:

$$red, blue, g_1(\alpha) \setminus \{red, blue\},$$
$$\{1, \ldots, k\} \setminus [g_1(\alpha) \cup \{orange, yellow\}], orange, yellow.$$

On the other hand, the order

$$blue, red, g_1(\alpha) \setminus \{red, blue\},$$
$$\{1, \ldots, k\} \setminus [g_1(\alpha) \cup \{orange, yellow\}], yellow, orange.$$

is used for $G_2 = G[V_2]$. Then, we obtain the following color sets (using our algorithm for the bipartite graph):

$g_1(x) = \{red\}$ $g_2(x) = \{blue\}$	$x \in X_1, c^{(1)}(x) = 1$ $x \in X_1, c^{(2)}(x) = 1$
$g_1(x) = \{yellow\}$ $g_2(x) = \{orange\}$	$x \in X_2, c^{(1)}(x) = 1$ $x \in X_2, c^{(2)}(x) = 1$
$yellow \notin g_1(x)$ $yellow \in g_2(x)$	$x \in X_1, c^{(1)}(x) = k-1$ $x \in X_1, c^{(2)}(x) = k-1$
$red \notin g_1(x)$ $red \in g_2(x)$	$x \in X_2, c^{(1)}(x) = k-1$ $x \in X_2, c^{(2)}(x) = k-1$

Case 3: The color sets $g_1(\alpha) = g_2(\alpha) = \{1, \ldots, k\}$ or $g_1(\alpha) = g_2(\alpha) = \emptyset$. In this case, we color the bipartite graph for G_1 in the order $1, 3, \ldots, k, 2$ and for G_2 in the order $2, \ldots, k, 1$. Again, we color the vertices in X_1 as early and vertices in X_2 as late as possible. Here, we obtain the following color sets:

$g_1(x) = \{1\}$ $g_2(x) = \{2\}$	$x \in X_1, c^{(1)}(x) = 1$ $x \in X_1, c^{(2)}(x) = 1$
$g_1(x) = \{2\}$ $g_2(x) = \{1\}$	$x \in X_2, c^{(1)}(x) = 1$ $x \in X_2, c^{(2)}(x) = 1$
$g_1(x) = \{1, 3, \ldots, k\}$ $g_2(x) = \{2, \ldots, k\}$	$x \in X_1, c^{(1)}(x) = k-1$ $x \in X_1, c^{(2)}(x) = k-1$
$g_1(x) = \{2, \ldots, k\}$ $g_2(x) = \{1, 3, \ldots, k\}$	$x \in X_2, c^{(1)}(x) = k-1$ $x \in X_2, c^{(2)}(x) = k-1$

We note that a case with $g_1(\alpha) = g_2(\alpha)$ and $c^{(1)}(\alpha) = c^{(2)}(\alpha) \in \{1, k-1\}$ is not possible (otherwise the invariants are not satisfied for α).

In all three cases, for each $x \in X_1 \cup X_2$ the color sets $g_i(x)$ satisfy the invariants (1) – (3). □

5 Example

An example of a parity graph $G = G^{(0)}$ and its generation using the three operations ϕ_1, ϕ_2 and ϕ_3 is given in Figure 9. In the next table we have illustrated the computation of the weights for the parity graph starting with $G^{(0)}$. We start with the orginal graph $G = G^{(0)}$ and weights $c(v) = 1$ for all $v \in V$. Notice that the size of a maximum clique in G is 4, and this size can be found in $G^{(4)}$ as the weight of a weighted clique $C = \{2, 3\}$ in the bipartite graph.

graph	1	2	3	4	5	6	7	8
$G^{(0)}$	1	1	1	1	1	1	1	1
$G^{(1)}$	1	1	1	1	1	1	1	-
$G^{(2)}$	1	1	1	1	1	1	-	-
$G^{(3)}$	1	2	1	1	1	-	-	-
$G^{(4)}$	1	2	2	1	-	-	-	-
$G^{(5)}$	1	-	-	-	-	-	-	-

Furthermore, in the next table, we have computed recursively two colorings for the graphs $G_1^{(0)}$ and $G_2^{(0)}$ using the three invariants. In this computation, we use 4 colors r, b, g and s and start with the single vertex graphs $G_1^{(5)}$ and $G_2^{(5)}$. Observe, that the color sets $g_1(v)$ and $g_2(v)$ of size 1 are different for each vertex v in $G_1^{(i)}$ and $G_2^{(i)}$ (for $0 \leq i \leq 5$). At the end, we obtain two colorings g_1, g_2 for $G_1^{(0)}$ and $G_2^{(0)}$ such that $g_1(x) \neq g_2(x)$ for all $x \in V$.

graph	1	2	3	4	5	6	7	8
$G_1^{(5)}$	$\{r\}$							
$G_2^{(5)}$	$\{b\}$							
$G_1^{(4)}$	$\{r\}$	$\{g,s\}$	$\{r,b\}$	$\{s\}$				
$G_2^{(4)}$	$\{b\}$	$\{r,s\}$	$\{b,g\}$	$\{r\}$				
$G_1^{(3)}$	$\{r\}$	$\{g,s\}$	$\{b\}$	$\{s\}$	$\{r\}$			
$G_2^{(3)}$	$\{b\}$	$\{r,s\}$	$\{g\}$	$\{r\}$	$\{b\}$			
$G_1^{(2)}$	$\{r\}$	$\{s\}$	$\{b\}$	$\{s\}$	$\{r\}$	$\{g\}$		
$G_2^{(2)}$	$\{b\}$	$\{r\}$	$\{g\}$	$\{r\}$	$\{b\}$	$\{s\}$		
$G_1^{(1)}$	$\{r\}$	$\{s\}$	$\{b\}$	$\{s\}$	$\{r\}$	$\{g\}$	$\{b\}$	
$G_2^{(1)}$	$\{b\}$	$\{r\}$	$\{g\}$	$\{r\}$	$\{b\}$	$\{s\}$	$\{g\}$	
$G_1^{(0)}$	$\{r\}$	$\{s\}$	$\{b\}$	$\{s\}$	$\{r\}$	$\{g\}$	$\{b\}$	$\{s\}$
$G_2^{(0)}$	$\{b\}$	$\{r\}$	$\{g\}$	$\{r\}$	$\{b\}$	$\{s\}$	$\{g\}$	$\{r\}$

6 Conclusions

In this paper, we have proved that the GOCCP problem restricted to parity graphs $G = (V, E)$ can be solved in polynomial time using a linear program if the costs $k_{v,c} = k_{v,3}$ for $c \geq 3$ and

$v \in V$. This result follows from the characterization that G is a parity graph, if and only if the cartesian product $G \times K_2$ is a perfect graph. Furthermore, we can show that the OCCP problem with three different cost values $k_1 = \ldots = k_q < k_{q+1} = \ldots = k_p < k_{p+1} = \ldots = k_n$ can be solved in polynomial time for bipartite graphs.

The following questions are interesting for further research:

(a) give a fast combinatorial algorithm for the GOCCP problem restricted to parity graphs and costs $k_{v,c} = k_{v,3}$ for $c \geq 3$,

(b) study the polyhedron for the GOCCP problem with $m = 3$ colors,

(c) find the complexity of the OCCP problem for parity graphs with three different cost values,

(d) study a modified integer linear program formulation for the OCCP problem with two and three different cost values.

References

[1] G.S. Adhar and S. Peng, Parallel algorithms for cographs and parity graphs with applications, *Journal of Algorithms* 11 (1990), 252-284.

[2] H.J. Bandelt and H.M. Mulder, Metric characterization of parity graphs, *Discrete Mathematics* 91 (1991), 221-230.

[3] M. Burlet and J.P. Uhry, Parity graphs, *Annals of Discrete Mathematics* 21 (1984), 253-277.

[4] V. Chvátal, On certain polytopes associated with graphs, *Journal Combinatorial Theory* B-18 (1975), 138–154.

[5] S. Cicerone and G. Di Stefano, On the equivalence in complexity among basic problems on bipartite graphs and parity graphs, to appear in: *International Symposium on Algorithms and Computation* ISAAC 97, Singapore, LNCS (1997).

[6] D. de Werra and A. Hertz, On perfectness of sums of graphs, Technical Report ORWP 87-13 and 97-06, Dept. de Mathématiques, Ecole Polytechnique Fédérale de Lausanne.

[7] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completness, Freeman, San Francisco, 1979.

[8] M. Grötschel, L. Lovász and A. Schrijver, Geometric Algorithms and Combinatorial Optimization, Springer, Berlin, 1988.

[9] K. Jansen, The optimum cost chromatic partition problem, *Algorithms and Complexity* CIAC 97, Rome, LNCS 1203 (1997), 25-36.

[10] L.G. Kroon, A. Sen, H. Deng and A. Roy, The optimal cost chromatic partition problem for trees and interval graphs, *Graph Theoretical Concepts in Computer Science* WG 96, Como, LNCS (1996).

[11] H. Meyniel, The graphs whose odd cycles have at least two crossing chords, *Annals of Discrete Mathematics* 21 (1984), 115-119.

[12] T. Przytycka and D.G. Corneil, Parallel algorithms for parity graphs, *Journal of Algorithms* 12 (1991), 96-109.

[13] G. Ravindra and K.R. Parathasarathy, Perfect product graphs, *Discrete Mathematics*, 20 (1977), 177-186.

[14] B. Reed, private communication.

[15] H. Sachs, On the Berge conjecture concerning perfect graphs, in: Combinatorial Structures and their Applications, Gordon and Beach, New York, 1970, 377-384.

[16] A. Sen, H. Deng and S. Guha, On a graph partition problem with an application to VLSI layout, *Information Processing Letters* 43 (1992), 87-94.

[17] K.J. Supowit, Finding a maximum planar subset of a set of nets in a channel, *IEEE Transactions on Computer Aided Design*, CAD 6, 1 (1987) 93–94.