Optim Lett (2018) 12:649–659
https://doi.org/10.1007/s11590-017-1217-7

ORIGINAL PAPER

Weighted matching with pair restrictions

Dorit S. Hochbaum[1] · **Asaf Levin**[2]

Received: 25 April 2016 / Accepted: 14 November 2017 / Published online: 22 November 2017

Abstract The weighted matroid parity problems for the matching matroid and gammoids are among the very few cases for which the weighted matroid parity problem is polynomial time solvable. In this work we extend these problems to a general revenue function for each pair, and show that the resulting problem is still solvable in polynomial time via a standard weighted matching algorithm. We show that in many other directions, extending our results further is impossible (unless P = NP). One consequence of the new polynomial time algorithm is that it demonstrates, for the first time, that a *prize-collecting assignment* problem with "pair restriction" is solved in polynomial time. The prize collecting assignment problem is a relaxation of the prize-collecting traveling salesman problem which requires, for any prescribed pair of nodes, either both nodes of the pair are matched or none of them are. It is shown that the prize collecting assignment problem is equivalent to the *prize collecting cycle cover* problem which is hence solvable in polynomial time as well.

Keywords Assignment · Traveling salesman problem · Prize collecting · Weighted graph matching · Matching matroid · Gammoid · Weighted matroid parity

Dorit S. Hochbaum research was supported in part by NSF Award No. CMMI-1200592.

✉ Asaf Levin
levinas@ie.technion.ac.il

Dorit S. Hochbaum
hochbaum@ieor.berkeley.edu

1 Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA 94720, USA

2 Faculty of Industrial Engineering and Management, The Technion, Haifa, Israel

1 Introduction

Let $G = (V, E)$ be an undirected graph, and let $S \subseteq V$ be a given subset of nodes. A *matching matroid* is defined over the ground set S where a subset $S' \subseteq S$ is independent in this matroid if there exists a matching in G that covers all nodes in S'. Note that for a matching matroid M, there is always a bipartite graph G' such that M is the matching matroid of G'.

A gammoid is a generalization of the matching matroid. Here, we consider a representation of a gammoid as used by Tong et al. [22]. A gammoid is represented by a bipartite graph $G = (V \cup U, E)$ (where V and U are the sides of the bipartition), and we are given a subset $V' \subseteq V$ such that G has a matching covering all nodes of $V \setminus V'$. The ground set of the gammoid is V', and a subset $I \subseteq V'$ is independent in the gammoid if there exists a matching covering all nodes in $I \cup (V \setminus V')$. Observe that if $V = V'$, then the gammoid is a matching matroid. This definition of a gammoid is based on its characterization as the contraction of a transversal matroid.

Given a matroid over a ground set S, the *weighted matroid parity* over this matroid is defined as follows. The elements of S are partitioned into pairs where each pair $\{s, s'\}$ is associated with a positive weight $\pi_{\{s,s'\}}$ and the objective is to find an independent set of the matroid consisting of a subset of pairs of maximum total weight (that is, an independent set that is a union of a subset of the pairs that maximizes the total weight of the pairs in this subset). The *unweighted matroid parity* is the special case of the weighted matroid parity where for every pair in the collection of pairs the weight of the pair is one.

The unweighted matroid parity was studied extensively (see e.g. Chapter 43 in [20] for a survey of results). Here, we mention that the problem is NP-hard for general matroids and unsolvable in polynomial time in the oracle model [20]. However, Lovász [14] established a polynomial time algorithm for the special case of the unweighted matroid parity for linear matroid with a given representation. The algorithm of Lovász as well as its later improvements [6,8,16,17] heavily use the fact that the problem is unweighted, and it is unclear how (or if possible at all) to generalize this result for the weighted case. The unweighted problem for general matroid has a polynomial time approximation scheme [13].

The complexity status of the weighted matroid parity is unclear. The unique cases for which polynomial time algorithms were established [22] are the gammoid and its special cases of the matching matroid and the transversal matroid. However, randomized pseudopolynomial time algorithms are known for the case of linear matroids with a given representation [5,15], and they lead to fully polynomial time randomized approximation schemes for this case of linear matroids with a given representation [18]. For general matroids, there is a 3/2-approximation algorithm [13], and for the special case of strongly base orderable matroids there is a polynomial time approximation scheme [21] (where a matroid is a strongly base orderable matroid if for any two bases B_1,B_2 of the matroid there exists a bijection $f : B_1 \to B_2$ such that for any $X \subseteq B_1$ the set $B_1 \setminus X \cup \bigcup_{x \in X} \{f(x)\}$ is also a base of the matroid).

In this work, we study the following generalization of the weighted matroid parity for gammoids and matching matroids that we call MATCHING WITH PAIRS ($\mathcal{MP}$). The input consists of an undirected (not necessarily bipartite) graph $G = (V, E)$ with weight for each edge where $w(e)$ denotes the weight of the edge e, a subset $V' \subseteq V$, and a set of pairs P where each pair is a subset of V with exactly two nodes, such that each node in V appears in at most one pair of P, for every $p \in P$, $p \subseteq V'$, and G has a matching that matches all nodes in $V \setminus V'$. Each pair $p \in P$ is associated with a reward function π^p defined over the power set of p (This means that there is a reward for the case that both nodes are matched, or one is matched, the other is matched, or none are matched). Both the weights of edges and the reward could be positive or negative. A feasible solution is a matching M in G that matches all nodes in $V \setminus V'$. For a given matching M, we denote by $V(M)$ the set of nodes that are matched in M, and $X(M)$ the set of nodes left exposed by M. The goal is to find a matching M in G such that $V \setminus V' \subseteq V(M)$ that maximizes the sum of the total weight of its edges and the total reward of all pairs in P, where a pair $p \in P$ is awarded a reward $\pi^p(V(M) \cap p)$. That is, the goal is to maximize the value of M defined as $Z(M) = \sum_{e \in M} w(e) + \sum_{p \in P} \pi^p(V(M) \cap p)$ over all matchings M of G such that $V \setminus V' \subseteq V(M)$. Note that the definition of the reward functions can be used to enforce logical conditions on pairs such as: for a given pair either both are matched or both are exposed (by setting the reward of other options to be $-\infty$), or for a pair $\{u, v\}$ if u is matched then v is also matched, etc. We let $V^P = \cup_{p \in P}\, p$ be the set of nodes that appear in pairs of P.

1.1 Motivation

A well known relaxation of the TSP defined on a graph $G = (V, E)$ is attained by relaxing the subtour elimination constraints. The relaxed problem is to find a cycle covering at minimum total edge cost (also known as the 2-factor problem). That is, a subset of edges $E' \subseteq E$ that forms a collection of cycles containing all nodes, and thus the degree of each node with respect to E' is 2 (or in the directed case each node has outdegree and indegree equal to 1 with respect to the set of arcs E'). The minimum cost cycle covering problem can be solved as an assignment problem and is thus polynomial time solvable. The construction of the assignment bipartite graph has the nodes of G duplicated, one copy on each side of the bipartition.

Consider next the *prize collecting TSP*, where the goal is to find a tour that visits a subset of the nodes so that the cost of the tour, minus the rewards/prizes collected from the nodes visited is minimum (see e.g. [1–4,7,10–12,19]). For the prize collecting TSP one might consider the analogous relaxation, the *prize-collecting assignment* problem, where the goal is to find a matching on a subset of the nodes, so that the cost of the matching minus the prize value associated with the subset of nodes is minimum. It is easy to see that this problem is solved in polynomial time as minimum weight bipartite matching.

Although the prize-collecting assignment problem is a relaxation of the prize-collecting TSP problem, it is a poor relaxation. That is, because the implied collection of arcs may have some nodes with indegree (outdegree) equal to 1 while the outdegree (indegree) is equal to 0. Therefore the resulting collection of edges may not form a collection of cycles. To impose the condition that in the prize collecting tour each node is of degree 2 or 0 one has to add *pair restrictions*. That is, if one node is matched then the other node of the pair has to be matched also. Thus either both nodes are matched, or neither of them is matched. With this restriction, the solution would be optimal for the *prize collecting cycle cover* in which the cost of the edges in the cycles, minus the sum of prizes derived from the nodes covered in the cycles, is minimum.

The prize-collecting assignment problem with *pair restrictions* is introduced here, for the first time, and is shown to be solved in polynomial time as a special case of $\mathcal{MP}$. More precisely, we define a *maximization* problem called PRIZE COLLECTING 2- FACTOR ($\mathcal{PCTF}$) (that is equivalent to the minimization problem discussed above) as follows. Given an input graph $G = (V, E)$ with cost $c(e)$ for each edge $e \in E$ and prize $\pi(v)$ for each node $v \in V$, the goal is to select a subset of nodes $S \subseteq V$ and an edge multi-set $E' \subseteq E$ where each edge $e \in E'$ connects two nodes of S (and each edge of E is allowed to be taken twice to E') and the degree of each node (of S) in the multi-graph (S, E') is exactly 2, so that $\sum_{v \in S} \pi(v) - \sum_{e \in E'} c(e)$ is maximized.

We now explain how to use the algorithm for $\mathcal{MP}$ for solving $\mathcal{PCTF}$: Given an input for $\mathcal{PCTF}$ with a graph $G = (U, E)$, edge costs $c(e)$, and node prizes $\pi(v)$, we construct a bipartite graph $G^{(2)} = (U \cup U', E^{(2)})$ as an input for $\mathcal{MP}$ by duplicating each node of $v \in U$ to two copies, v, v' (where $v \in U$ and $v' \in U'$). For each edge $e = [u, v] \in E$ there are two edges $[u, v'], [u', v] \in E^{(2)}$ both of weight $w([u, v']) = w([u', v]) = -c(e)$. Let the list of pairs be $\{\{u, u'\} : u \in U\}$ (that is, for each original node we have a pair consisting of its two copies). The input for $\mathcal{MP}$ is defined by letting $\pi^{\{u,u'\}}(A)$ be $\pi(u)$ if $A = \{u, u'\}$, 0 if $A = \emptyset$, and $-\infty$ otherwise (meaning we are not allowed to match only one copy of the original node u). In the resulting instance, every solution of positive value has the property that for each original node $u \in U$ either both copies are matched (in this case we will say that u is in the solution) or both are exposed. Based on an optimal solution for the instance of $\mathcal{MP}$ we will create an optimal solution for the prize collecting 2-factor instance by defining S as the set of nodes in the solution and E' as the set of edges for which at least one of the copies of the edge is in the solution (if both copies are in the solution for $\mathcal{MP}$, we will take two copies of the edge for E').

1.1.1 Paper outline

In Sect. 2, we show that $\mathcal{MP}$ is polynomially solvable using an algorithm for computing a maximum weight matching in an auxiliary multi-graph with $O(|V| + |P|)$ nodes and $O(|E| + |P|)$ edges. This results in a polynomial time algorithm for solving $\mathcal{PCTF}$. In Sect. 4 we show that various generalizations of the result of Sect. 2 are impossible as these generalizations are NP-hard in the strong sense.

2 Solving $\mathcal{MP}$ in polynomial time using a maximum weight matching

Given the input to $\mathcal{MP}$ we construct a new (multi-)graph $\bar{G} = (\bar{V}, \bar{E})$ where G is a subgraph of $\bar{G}$, by adding a gadget consisting of three new nodes and five new edges for each pair. Formally, $\bar{V} = V \cup \cup_{p \in P} V_p$ and $\bar{E} = E \cup \cup_{p \in P} E_p$ where in $\bar{E}$ there might be parallel edges. For every $p \in P$, V_p consists of three new nodes (distinct for each pair in P) denoted as $q_1^{(p)}, q_2^{(p)}, q_3^{(p)}$. Next, we define the edges in E_p for a pair $p = \{u, v\}$

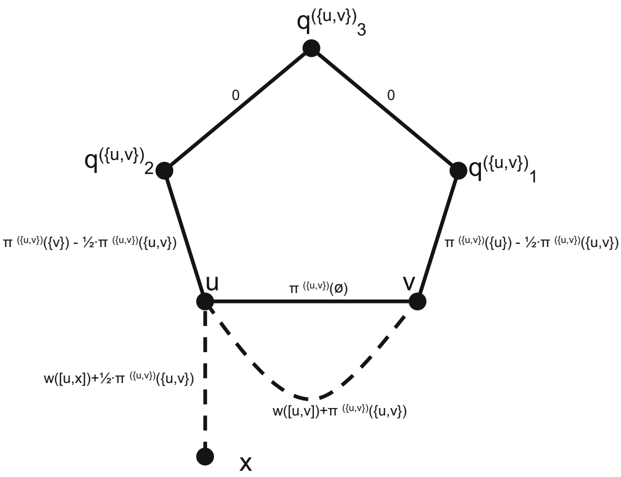

Fig. 1 An example of the graph $\bar{G}$ where G has three nodes x, u, v and two edges $[x, u]$, $[u, v]$ and P consists of one pair $\{u, v\}$. The dashed edges are edges in E whereas the solid edges are in $E_{\{u,v\}}$. Next to each edge we state its weight according to w'

$(u \neq v)$. We have $E_p = \left\{\left[v, q_1^{(p)}\right], \left[u, q_2^{(p)}\right], [u, v], \left[q_1^{(p)}, q_3^{(p)}\right], \left[q_2^{(p)}, q_3^{(p)}\right]\right\}$, thus $(p \cup V_p, E_p)$ is a cycle over five nodes $u, v, q_1^{(p)}, q_3^{(p)}, q_2^{(p)}$ appearing along the cycle in this order.

Finally, we define the new weight function $w' : \bar{E} \to R$ defined over the edges of $\bar{G}$. First, consider an edge $[u, v] \in E$. If $u, v \notin V^P$, then $w'([u, v]) = w([u, v])$; If $u \in V^P$ and $v \notin V^P$, let $x \in V$ be such that $\{u, x\} \in P$, and define $w'([u, v]) = w([u, v]) + \frac{\pi^{\{u,x\}}(\{u,x\})}{2}$; similarly, if $u \notin V^P$ and $v \in V^P$, let $x \in V$ be such that $\{v, x\} \in P$, and define $w'([u, v]) = w([u, v]) + \frac{\pi^{\{v,x\}}(\{v,x\})}{2}$; if $\{u, v\} \in P$, then define $w'([u, v]) = w([u, v]) + \pi^{\{u,v\}}(\{u, v\})$; last, if $u, v \in V^P$ and $\{u, v\} \notin P$, let $x, y \in V$ be such that $\{u, x\}, \{v, y\} \in P$ and define $w'([u, v]) = w([u, v]) + \frac{\pi^{\{u,x\}}(\{u,x\})}{2} + \frac{\pi^{\{v,y\}}(\{v,y\})}{2}$. This completes the description of the new weights of edges in E.

Next, consider an edge of E_p for a pair $p = \{u, v\}$. We define the following weights of the five edges in E_p. $w'([u, v]) = \pi^{\{u,v\}}(\emptyset)$, $w'\left(\left[v, q_1^{(p)}\right]\right) = \pi^{\{u,v\}}(\{u\}) - \frac{\pi^{\{u,v\}}(\{u,v\})}{2}$, $w'\left(\left[u, q_2^{(p)}\right]\right) = \pi^{\{u,v\}}(\{v\}) - \frac{\pi^{\{u,v\}}(\{u,v\})}{2}$, $w'\left(\left[q_1^{(p)}, q_3^{(p)}\right]\right) = w'\left(\left[q_2^{(p)}, q_3^{(p)}\right]\right) = 0$. We refer to Fig. 1 for an illustration of the gadget with the new weight function.

In this resulting graph $\bar{G}$ with resulting edge weight w' we compute a maximum weight matching M' such that $(V \setminus V') \cup \cup_{p\in P} \left(p \cup \left\{q_3^{(p)}\right\}\right) \subseteq V(M')$ (so we compute a maximum weight matching under the constraint that this set of nodes must be matched).

Lemma 1 *We have the following properties.*

1. *Given a matching M in G that is a feasible solution for $\mathcal{MP}$, there exists a matching M' in $\bar{G}$ that is feasible for the auxiliary problem defined above, such that the total weight of M' according to w' equals the value of M as a solution to $\mathcal{MP}$.*
2. *Given a matching M' in $\bar{G}$ that is a feasible for the auxiliary problem defined above, there exists a matching $M = M' \cap E$ in G that is feasible for $\mathcal{MP}$, such that the total weight of M' according to w' equals the value of M as a solution to $\mathcal{MP}$.*

Proof Computing the matching M' from M, we start with $M' = M$, and process the pairs in P one by one. Each such pair p will have a corresponding iteration, in which we add edges from E_p to the matching. Consider a pair $p = \{u, v\}$ that is considered in the current iteration.

If $p \subseteq X(M)$, then we add the edges $[u, v]$, $\left[q_1^{(p)}, q_3^{(p)}\right]$ to (the current) matching M'. If $p \cap V(M) = \{u\}$, then we add the edges $\left[v, q_1^{(p)}\right]$, $\left[q_2^{(p)}, q_3^{(p)}\right]$ to the matching M'. Similarly, if $p \cap V(M) = \{v\}$, then we add the edges $\left[u, q_2^{(p)}\right]$, $\left[q_1^{(p)}, q_3^{(p)}\right]$ to the matching M'. Finally, if $p \subseteq V(M)$, then we add to the matching the edge $\left[q_1^{(p)}, q_3^{(p)}\right]$.

Observe that when we process the pair $p = \{u, v\}$ we added to the matching only edges from E_p in a way that the resulting set M' is indeed a matching, and since we only augment M, the resulting matching matches all nodes in $V \setminus V'$. The feasibility of the matching is ensured by the fact that whenever we process a pair p, in all cases the nodes in $p \cup \{q_3^{(p)}\}$ are matched either in M or by the new edges we add to M' in the current iteration. This shows that if M is feasible for the $\mathcal{MP}$ instance, then the constructed matching M' is feasible for the auxiliary problem.

In the other direction, if M' is feasible for the auxiliary problem, then all nodes of $V \setminus V'$ are matched in M'. Observe that in $\bar{G}$ all edges adjacent to nodes of $V \setminus V'$ are edges of E (because $V^P \subseteq V'$) and thus $M = M' \cap E$ is a feasible solution for $\mathcal{MP}$. Thus, we conclude that M is a feasible solution for $\mathcal{MP}$ if and only if M' is feasible for the auxiliary problem.

It remains to show that $Z(M)$ (the value of M as a solution to $\mathcal{MP}$) equals the total weight of M'. To do that, we split the weight according to w' of edges of M incident to nodes in V^P; a weight of such an edge $[x, y]$ is split into an original weight of $w([x, y])$, a prize value of $\frac{\pi^{\{x,z\}}(\{x,z\})}{2}$ for x if there exists z such that $\{x, z\} \in P$, and a prize value of $\frac{\pi^{\{y,z\}}(\{y,z\})}{2}$ for y if there exists z such that $\{y, z\} \in P$ (this means in particular that if $\{x, y\} \in P$ then we split $w'([x, y]) - w([x, y])$ evenly between x and y, and otherwise if $|\{x, y\} \cap V^P| = 1$, then $w'([x, y]) - w([x, y])$ is the prize value of the node of $\{x, y\} \cap V^P$). For an edge in E_p, we say that its weight according to w' is a prize value. Now, the claim that $Z(M)$ equals the total weight of M' follows by showing that for every pair $p = \{u, v\} \in P$ in all cases the total prize values of the edges in E_p and the nodes in p equals the reward $\pi^p(V(M) \cap p)$ of p (in M).

To prove the last claim consider a pair $p = \{u, v\} \in P$. First, consider the direction in which we augment the matching M into a feasible solution for the auxiliary problem. If $p \cap V(M) = \emptyset$, then the matching is augmented by the edges $[u, v]$, $\left[q_1^{(p)}, q_3^{(p)}\right]$

of total weight of $\pi^{\{u,v\}}(\emptyset) + 0$ that equals the reward of p. If $p \cap V(M) = \{u\}$, then the matching is augmented by the edges $\left[v, q_1^{(p)}\right], \left[q_2^{(p)}, q_3^{(p)}\right]$ of total weight of $\pi^{\{u,v\}}(\{u\}) - \frac{\pi^{\{u,v\}}(\{u,v\})}{2}$ and together with the prize value of u, we get a total prize value that equals the reward of p. Similarly, if $p \cap V(M) = \{v\}$, then the matching is augmented by the edges $\left[u, q_2^{(p)}\right], \left[q_1^{(p)}, q_3^{(p)}\right]$ of total weight of $\pi^{\{u,v\}}(\{v\}) - \frac{\pi^{\{u,v\}}(\{u,v\})}{2}$ and together with the prize value of v, we get a total prize value that equals the reward of p. Finally, if $p \subseteq V(M)$, then we add to the matching the edge $\left[q_1^{(p)}, q_3^{(p)}\right]$ of zero weight, but each of u and v obtains the prize value $\frac{\pi^{\{u,v\}}(\{u,v\})}{2}$, and together this equals the reward of p.

Finally, consider the direction in which we are given a feasible solution for the auxiliary problem M' and we define the matching $M = M' \cap E$ as a feasible solution for $\mathcal{MP}$. We will show that the total weight of $M' \cap E_p$ together with the prize values of u and v equals the reward of p (according to M). If $p \cap V(M) = \emptyset$, then $M' \cap E_p$ must be either $[u, v], \left[q_1^{(p)}, q_3^{(p)}\right]$ or $[u, v], \left[q_2^{(p)}, q_3^{(p)}\right]$ (because no other feasible matching in $\left(\left\{u, v, q_1^{(p)}, q_2^{(p)}, q_3^{(p)}\right\}, E_p\right)$ covers $u, v, q_3^{(p)}$, and in both cases the total weight of the edges in $M' \cap E_p$ is $\pi^{\{u,v\}}(\emptyset) + 0$ that equals the reward of p. If $p \cap V(M) = \{u\}$, then $M' \cap E_p = \left\{\left[v, q_1^{(p)}\right], \left[q_2^{(p)}, q_3^{(p)}\right]\right\}$ of total weight of $\pi^{\{u,v\}}(\{u\}) - \frac{\pi^{\{u,v\}}(\{u,v\})}{2}$ and together with the prize value of u, we get a total prize value that equals the reward of p. Similarly, if $p \cap V(M) = \{v\}$, then $M' \cap E_p = \left\{\left[u, q_2^{(p)}\right], \left[q_1^{(p)}, q_3^{(p)}\right]\right\}$ of total weight of $\pi^{\{u,v\}}(\{v\}) - \frac{\pi^{\{u,v\}}(\{u,v\})}{2}$ and together with the prize value of v, we get a total prize value that equals the reward of p. Finally, if $p \subseteq V(M)$, then $M' \cap E_p$ is either $\left[q_1^{(p)}, q_3^{(p)}\right]$ or $\left[q_2^{(p)}, q_3^{(p)}\right]$ of zero weight, but each of u and v obtains the prize value $\frac{\pi^{\{u,v\}}(\{u,v\})}{2}$, and together this equals the reward of p. □

Based on Lemma 1, an algorithm that constructs an instance for the auxiliary problem, finds a maximum weight matching satisfying the conditions of that instance, and defines a solution to $\mathcal{MP}$ by identifying the set of edges from the solution to the auxiliary problem belonging to E, solves $\mathcal{MP}$ optimally in the same time complexity as the algorithm for maximum weighted matching in the auxiliary graph (that must match a given subset of nodes). Thus, we conclude our main result stated as follows.

Theorem 2 *Problem $\mathcal{MP}$ can be solved in polynomial time using an algorithm for maximum weighted matching.*

3 Applying the algorithm for solving $\mathcal{PCTF}$

For the instance of $\mathcal{PCTF}$ we are guaranteed that there exists a solution of finite value (i.e., a solution that does not use edges of $\bar{G}$ of weight $-\infty$. Thus, in the instance of the auxiliary problem that we create (after transforming the instance of $\mathcal{PCTF}$ to an instance G of $\mathcal{MP}$) for a pair $p = \{u, u'\}$ we never use the edges $\left[u', q_1^{(p)}\right], \left[u, q_2^{(p)}\right]$ as they have weight of $-\infty$, and thus we can delete them from the graph $\bar{G}$. In the

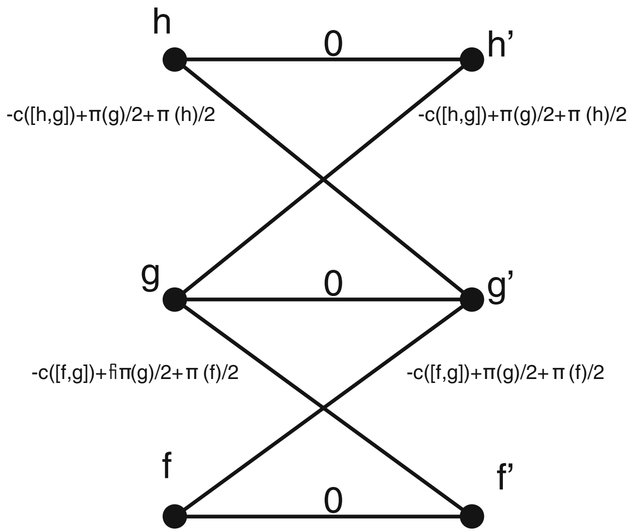

Fig. 2 An example of the resulting graph for solving $\mathcal{PCTF}$ via maximum weight bipartite matching where G is a path over three nodes f, g, h. Next to each edge we state its weight according to w'

resulting graph, for every pair we have a connected component of the three additional nodes of its gadget, and picking one edge of every such connected component will match the nodes in $\left\{q_3^{(p)} : p \in P\right\}$, and thus we can consider the problem of the remaining nodes.

Thus, we can remove from $\bar{G}$ all nodes of the form $q_1^{(p)}, q_2^{(p)}, q_3^{(p)}$ for all pairs p. In the resulting graph the auxiliary problem asks to match all nodes where beside the edges of G with their modified weight w' (such that $w'([u, v']) = -c([u, v]) + \pi(u)/2 + \pi(v)/2$) we add the zero weight edges $[u, u']$ for every node in the instance of $\mathcal{PCTF}$. See Fig. 2 for an illustration of the resulting graph and the weight of edges (according to w'). The resulting graph is a bipartite graph, and we need to compute a maximum weight perfect matching in it in order to solve optimally the auxiliary problem. Thus, we conclude the following.

Proposition 3 *There exists an algorithm for solving $\mathcal{PCTF}$ on a graph with n nodes and m edges, in the same time complexity of computing a maximum weight perfect matching in a bipartite graph with $2n$ nodes and $2m + n$ edges.*

Remark 4 Given an instance of weighted matroid parity for gammoids, one can define a reward function for every pair of nodes (in the set of pairs) $p = \{s, s'\}$ in the following way $\pi^p(\{s\}) = \pi^p(\{s'\}) = -\infty$, $\pi^p(\emptyset) = 0$, and $\pi^p(p)$ is the weight of this pair in the weighted matroid parity problem. Then, we can apply the same transformation as we did above for $\mathcal{PCTF}$ and obtain a graph where we look for a maximum weight matching. This graph that we obtained is equivalent to the instance of the maximum weight matching instance of Tong et al. [22]. However, our general construction for $\mathcal{MP}$ differs from this construction of Tong et al. [22].

4 NP-hard generalizations of $\mathcal{MP}$

We first show that generalizing $\mathcal{MP}$ to disjoint set of triples makes the problem NP-hard. Formally, we define the following optimization problem that we call MATCHING WITH TRIPLES ($\mathcal{MT}$). The input consists of an undirected graph $G = (V, E)$, and a set of triples P where each triple is a subset of V consisting of exactly three nodes, such that each node in V appears in at most one triple of P. The goal is to find a matching M in G that maximizes the number of triples in P for which M matches all members of the triple, i.e., to maximize $|\{p \in P : p \subseteq V(M)\}|$. Notice that this is a special case of the reward function that assigns non-zero reward only if all nodes of the triple are matched. In particular $\mathcal{MT}$ generalizes the unweighted matroid parity for the matching matroid to the case of subsets of three elements in each instead of subsets of two elements (as in the matroid parity problem).

Theorem 5 *Problem $\mathcal{MT}$ is NP-hard in the strong sense.*

Proof We show a reduction from Exact 3-Cover problem (X3C) defined as follows. The input for X3C is a ground set $S = \{1, 2, \ldots, 3n\}$ and a collection $S_1, S_2, \ldots, S_m$ of subsets of the ground set where $|S_i| = 3$ for all i. The goal is to decide if there is a sub-collection of the subsets such that each element of the ground set appears in exactly one of the subsets in the sub-collection. X3C is NP-hard in the strong sense (see [9]).

Given an input for X3C, we construct a bipartite graph $G = (V, E)$ where the nodes in V are partitioned into element nodes and subsets nodes. There is one element node in G for each element of the ground set, and there are three subsets nodes for each subset in the collection of subsets. With a slight abuse of notation, for an element e, we denote by e both the element and its element node. For a subset S_i in the collection of subsets, we denote by S_i^1, S_i^2, S_i^3 its three nodes. The set of triples is $\left\{p = \left\{S_i^1, S_i^2, S_i^3\right\} : i = 1, 2, \ldots, m\right\}$, and it remains to describe the edge set of the graph G. For each subset $S_i = \{a_i, b_i, c_i\}$ in the collection of subsets such that $a_i < b_i < c_i$ we have three edges in G as follows: $\left[a_i, S_i^1\right], \left[b_i, S_i^2\right], \left[c_i, S_i^3\right]$. This completes the description of the input for $\mathcal{MT}$.

To prove the theorem, it suffices to show that there is a solution for X3C if and only if the optimal solution for the constructed input for $\mathcal{MT}$ is at least n. Given a matching M in G of value (at least) n, for a triple $p = \left\{S_i^1, S_i^2, S_i^3\right\}$ of our collection of triples, we have $\left\{S_i^1, S_i^2, S_i^3\right\} \subseteq V(M)$ if and only if we choose S_i to the collection of subsets in the constructed solution for X3C. Now, since the elements nodes have degree at most 1 in the matching, each element appears in at most one chosen subset, and since the matching has value at least n we chose at least n subsets, and thus by counting, each element appears in exactly one chosen subsets. That is, the instance for X3C is feasible.

In the other direction, assume that there is a solution for X3C, then we start with an empty matching M, and for every chosen subset $S_i = \{a_i, b_i, c_i\}$ (to the solution of X3C), we add to the matching M the three edges $[a_i, S_i^1]$, $[b_i, S_i^2]$, $[c_i, S_i^3]$. Then, clearly the triple $\left\{S_i^1, S_i^2, S_i^3\right\}$ satisfies $\left\{S_i^1, S_i^2, S_i^3\right\} \subseteq V(M)$, and since each element of the ground set appears in at most one chosen subset, the constructed set of edges is indeed a matching in G. Thus, the constructed matching M has value at least n as a solution for $\mathcal{MT}$. □

We next show that the variant of $\mathcal{MP}$ where pairs are defined as pair of edges and the objective is to maximize the number of pairs belonging to a matching is NP-hard (in contrast to $\mathcal{MP}$ where the pairs are pairs of nodes and the problem is polynomially solvable). More precisely, we define the following optimization problem that we call MATCHING WITH EDGE PAIRS ($\mathcal{MEP}$). The input consists of an undirected multi-graph $G = (V, E)$, and a set of pairs P where each pair is a subset of E consisting of exactly two edges, such that each edge in E appears in at most one pair of P. The goal is to find a matching M in G that maximizes the number of pairs in P for which M contains the pair, i.e., to maximize $|\{p \in P : p \subseteq M\}|$.

Theorem 6 *Problem $\mathcal{MEP}$ is NP-hard in the strong sense.*

Proof We show a reduction from 3-Dimensional Matching problem (3DM) defined as follows. We are given three disjoint sets $A = \{a_1, a_2, \ldots, a_n\}$, $B = \{b_1, b_2, \ldots, b_n\}$, and $C = \{c_1, c_2, \ldots, c_n\}$ each has n elements, and a collection of subsets $T = \{T_1, T_2, \ldots, T_m\}$ where each T_i has exactly one element of each of the sets A, B, C. The goal is to find out if there exists a sub-collection T' of T, such that each element of $A \cup B \cup C$ appears in exactly one subset of T'. 3DM is NP-complete in the strong sense (see [9]).

Given an instance for 3DM, we construct an instance for $\mathcal{MEP}$ as follows. The node set of the multi-graph is $A \cup B \cup C \cup T$, and the edge set is defined as follows. For each $T_i = \{\bar{a}, \bar{b}, \bar{c}\}$ we have the two edges $[\bar{a}, \bar{b}]$ and $[\bar{c}, T_i]$ in E, and we have a pair consisting of these two edges. Observe that this definition allows for parallel edges if $\bar{a}$ and $\bar{b}$ belong to more than one common subset. We say that these two edges correspond to T_i. We claim that there is a feasible solution for 3DM if and only if there is a solution M for $\mathcal{MEP}$ of value at least n.

If there is a feasible solution T' for 3DM, then we let M be the matching consisting of the edges $\{[\bar{a}, \bar{b}], [\bar{c}, T_i] : T_i = \{\bar{a}, \bar{b}, \bar{c}\} \in T'\}$ (that is, the matching edges are the edges that correspond to the subsets in the solution for 3DM). Then, the value of this collection of edges is clearly $|T'| = n$, and this is a matching because each node of T appears in at most one edge in G, while every node in $A \cup B \cup C$ appears at most once in the selected subsets of T' and thus have degree at most one in the subgraph whose edge set is the selected edges.

In the other direction, let M be a matching of value at least n as a solution for $\mathcal{MEP}$ in the multi-graph G, then we create a sub-collection of subsets T' by choosing to T' all subsets whose both corresponding edges belong to M. Thus, we choose a sub-collection of at least n subsets, and it suffices to show that each element appears at most once in the sub-collection of chosen subsets. This last claim holds because if we assume by contradiction that there is an element e that appears in (at least) two chosen subsets, then its node in the multi-graph has (at least) two incident edges in the matching, and this is a contradiction. □

Acknowledgements The topic of prize collecting 2-factor arose from a discussion the first author had with Alejandro Toriello concerning relaxations for the prize collecting traveling salesman problem.

References

1. Archetti, C., Speranza, M.G., Vigo, D.: Vehicle routing problems with profits. In: Vehicle Routing: Problems, Methods, and Applications, SIAM, pp. 273–298 (2014)
2. Balas, E.: The prize collecting traveling salesman problem. Networks **19**, 621636 (1989)
3. Balas, E.: The prize collecting traveling salesman problem and its applications. In: The Traveling Salesman Problem and Its Variations, Springer, pp. 663–695 (2007)
4. Bienstock, D., Goemans, M.X., Simchi-Levi, D., Williamson, D.P.: A note on the prize collecting traveling salesman problem. Math. Program. **59**, 413–420 (1993)
5. Camerini, P., Galbiati, G., Maffioli, F.: Random pseudo-polynomial algorithms for exact matroid problems. J. Algorithms **13**, 258–273 (1992)
6. Cheung, H.Y., Lau, L.C., Leung, K.M.: Algebraic algorithms for linear matroid parity problems. ACM Trans. Algorithms (TALG), **10**(3), article number 10 (2014)
7. Feillet, D., Dejax, P., Gendreau, M.: Traveling salesman problems with profits. Transp. Sci. **39**(2), 188–205 (2005)
8. Gabow, H.N., Stallmann, M.: An augmenting path algorithm for linear matroid parity. Combinatorica **6**, 123–150 (1986)
9. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco (1979)
10. Goemans, M.X., Williamson, D.P.: A general approximation technique for constrained forest problems. SIAM J. Comput. **24**, 296–317 (1995)
11. Goemans, M.X.: Combining approximation algorithms for the prize-collecting TSP. arXiv:0910.0553v1 (2009)
12. Hoffman, K.L., Padberg, M., Rinaldi, G.: Traveling salesman problem. In: Encyclopedia of Operations Research and Management Science, Springer, pp. 1573–1578 (2013)
13. Lee, J., Sviridenko, M., Vondrak, J.: Matroid matching: the power of local search. SIAM J. Comput. **42**, 357–379 (2013)
14. Lovász, L.: The matroid matching problem. In: Algebraic Methods in Graph Theory, pp. 495–517, Szeged (1978)
15. Narayanan, H., Saran, H., Vazirani, V.: Randomized parallel algorithms for matroid union and intersection, with applications to arboresences and edge-disjoint spanning trees. SIAM J. Comput. **23**, 387–397 (1994)
16. Orlin, J.B.: A fast, simpler algorithm for the matroid parity problem. Proc. IPCO **240–258**, 2008 (2008)
17. Orlin, J.B., Vate, J.H.V.: Solving the linear matroid parity problem as a sequence of matroid intersection problems. Math. Program. **47**, 81–106 (1990)
18. Promel, H.J., Steger, A.: A new approximation algorithm for the Steiner tree problem with performance ratio 5/3. J. Algorithms **36**, 89–101 (2000)
19. Riera-Ledesma, J., Salazar-Gonzalez, J.-J.: Solving school bus routing using the multiple vehicle traveling purchaser problem: a branch-and-cut approach. Comput. Oper. Res. **39**(2), 391–404 (2012)
20. Schrijver, A.: Combinatorial Optimization Polyhedra and Efficiency. Springer, Berlin (2003)
21. Soto, A.J.: A simple PTAS for weighted matroid matching on strongly base orderable matroids. Discrete Appl. Math. **164**, 406–412 (2014)
22. Tong, P., Lawler, E.L., Vazirani, V.V.: Solving the Weighted Parity Problem for Gammoids by Reduction to Graphic Matching. University of California, Oakland (1982)