Budgeted Matching via the Gasoline Puzzle*

Guido Schäfer

Abstract We consider a natural generalization of the classical matching problem: In the *budgeted matching problem* we are given an undirected graph with egde weights, non-negative edge costs and a budget. The goal is to compute a matching of maximum weight such that its cost does not exceed the budget. This problem is weakly NP-hard. We present the first polynomial-time approximation scheme for this problem. Our scheme computes two solutions to the Lagrangian relaxation of the problem and patches them together to obtain a near-optimal solution. In our patching procedure we crucially exploit the adjacency relations of vertices of the matching polytope and the solution to an old combinatorial puzzle.

1 Problem definition

The *budgeted matching problem* is a natural generalization of the classical matching problem: We are given an undirected graph $G = (V, E)$ with edge weights $w : E \to \mathbb{Q}$, non-negative edge costs $c : E \to \mathbb{Q}^+$ and a budget $B \in \mathbb{Q}^+$. Recall that a matching of G is a subset $M \subseteq E$ of the edges such that no two edges of M share a common node. Let $\mathcal{F}$ be the set of all matchings of G. Define the weight of a matching M as the total weight of all edges in M, i.e., $w(M) := \sum_{e \in M} w(e)$. Similarly, the cost of M is defined as $c(M) := \sum_{e \in M} c(e)$. The goal is to compute a matching of maximum weight whose cost is at most B, i.e.,

$$\text{maximize } w(M) \;\text{ subject to }\; M \in \mathcal{F},\; c(M) \le B. \qquad (\bar{\Pi})$$

Guido Schäfer
Centrum Wiskunde & Informatica, Science Park 123, 1098 XG Amsterdam, The Netherlands, and
VU University Amsterdam, De Boelelaan 1105, 1081 HV Amsterdam, The Netherlands,
e-mail: g.schaefer@cwi.nl.

* The exposition of the results given here is based on the article [1].

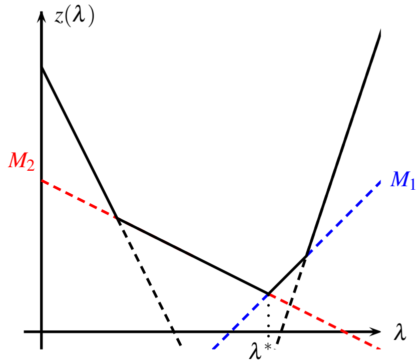

Fig. 1 The Lagrangian value $z(\lambda)$ as a function of λ (solid line). Each dashed line represents the Lagrangian value of a specific solution.

The budgeted matching problem is weakly NP-hard even for bipartite graphs. This follows by a simple reduction from the knapsack problem. Here we present the first polynomial-time approximation scheme (PTAS) for the budgeted matching problem. For a given $\varepsilon > 0$, our algorithm computes a $(1-\varepsilon)$-approximate solution to the problem in time $O(m^{O(1/\varepsilon)})$, where m is the number of edges in the graph.

Subsequently, we use OPT to refer to the weight of an optimal solution M^* to $(\bar{\Pi})$. Also, we use (Π) to refer to the respective unbudgeted matching problem (where the budget constraint "$c(M) \leq B$" is dropped).

2 A PTAS for Budgeted Matching

Consider the Lagrangian relaxation $\mathrm{LR}(\lambda)$ of the budgeted matching problem $(\bar{\Pi})$:

$$z(\lambda) := \text{maximize } \big(w(M) + \lambda(B - c(M))\big) \text{ subject to } M \in \mathcal{F}. \qquad (\mathrm{LR}(\lambda))$$

Note that every feasible solution to the budgeted matching problem $(\bar{\Pi})$ satisfies $c(M) \leq B$. Thus, for every $\lambda \geq 0$ the optimal solution to $\mathrm{LR}(\lambda)$ gives an upper bound on OPT, i.e., $z(\lambda) \geq \mathrm{OPT}$. The Lagrangian dual problem is to find the best such upper bound, i.e., to determine λ^* such that $z(\lambda^*) = \min_{\lambda \geq 0} z(\lambda)$ (see also Figure 1).

Note that for a fixed value of λ the Lagrangian relaxation $\mathrm{LR}(\lambda)$ is equivalent to solving a maximum weight matching problem with respect to the *Lagrangian weights*

$$w_\lambda(e) := w(e) - \lambda c(e) \qquad \forall e \in E.$$

Given that there are combinatorial algorithms to solve the maximum weight matching problem, we can use standard parametric search techniques (see Section 4 for references) to determine an optimal Lagrangian multiplier λ^* in strongly polynomial time. In addition, we can compute within the same time bound two optimal matchings M_1 and M_2 to $\mathrm{LR}(\lambda^*)$ such that $c(M_1) \leq B \leq c(M_2)$.

The idea now is to *patch* M_1 and M_2 together to obtain a feasible solution M to $(\bar{\Pi})$ whose weight $w(M)$ is not too far from the optimal one. More precisely, the following lemma will be crucial to derive our polynomial-time approximation scheme:

Lemma 1 (Patching Lemma). *There is a polynomial-time algorithm to compute a solution M to the budgeted matching problem of weight $w(M) \geq \mathrm{OPT} - 2w_{\max}$, where $w_{\max}$ is the largest weight of an edge.*

A formal proof of this lemma is given in Section 3. Intuitively, our patching procedure consists of two phases: an *exchange phase* and an *augmentation phase*.

Exchange Phase: Consider the polytope induced by the set of feasible matchings $\mathcal{F}$ and let F be the face given by the solutions of maximum Lagrangian weight w_{λ^*}. This face contains both M_1 and M_2. We now iteratively replace either M_1 or M_2 with another vertex on F, preserving the invariant $c(M_1) \leq B \leq c(M_2)$, until M_1 and M_2 correspond to adjacent vertices of the matching polytope. Note that the Lagrangian weight of M_i, $i \in \{1,2\}$, is $w_{\lambda^*}(M_i) = z(\lambda^*) \geq \mathrm{OPT}$. However, with respect to the original weight, we can only infer that $w(M_i) = z(\lambda^*) - \lambda^*(B - c(M_i))$. That is, we cannot hope to use these matchings directly: M_1 is a feasible solution to $(\bar{\Pi})$ but its weight $w(M_1)$ might be arbitrarily far from OPT. In contrast, M_2 has weight $w(M_2) \geq \mathrm{OPT}$, but is infeasible.

Augmentation Phase: In order to overcome the above problem, we exploit the adjacency relation between M_1 and M_2. It is known that two matchings M_1 and M_2 are adjacent in the matching polytope if and only if their symmetric difference $X = M_1 \oplus M_2$ is an alternating cycle or a path. The idea now is to patch M_1 according to a properly chosen subpath X' of X. We ensure that the subpath X' is chosen such that the Lagrangian weight of M_1 does not decrease too much, while at the same time the gap between the budget B and the cost of M_1 (and hence also the gap between $w(M_1)$ and $z(\lambda^*)$) is reduced. This way we obtain a feasible solution M whose weight differs from OPT by at most $2w_{\max}$.

Surprisingly, our proof that such a patching subpath X' always exists is based on the solution of an old combinatorial puzzle, also known as the *Gasoline Puzzle*:

> "Along a speed track there are some gas-stations. The total amount of gasoline available in them is equal to what our car (which has a very large tank) needs for going around the track. Prove that there is a gas-station such that if we start there with an empty tank, we shall be able to go around the track without running out of gasoline."

With the help of our Patching Lemma we derive a polynomial-time approximation scheme by "guessing" the $\Theta(1/\varepsilon)$ largest weight edges in an optimum solution.

Theorem 1. *There is a deterministic algorithm that for every $\varepsilon > 0$ computes a solution to the budgeted matching problem of weight at least $(1-\varepsilon)\mathrm{OPT}$ in time $O(m^{2/\varepsilon+O(1)})$, where m is the number of edges in the graph.*

Proof. Let $\varepsilon \in (0,1)$ be a given constant. Assume that the optimum matching M^* contains at least $p := \lceil 2/\varepsilon \rceil$ edges. (Otherwise the problem can be solved optimally by complete enumeration.)

Consider the following algorithm: First, we guess the p largest weight edges M^*_H of M^*. We then remove from the graph G the edges in M^*_H, all edges incident to M^*_H, and all edges of weight larger than the smallest weight in M^*_H. We also decrease the budget by $c(M^*_H)$. Let I' be the resulting budgeted matching instance. Note that the maximum weight of an edge in I' is

$$w'_{\max} \le \tfrac{1}{p} w(M^*_H) \le \tfrac{1}{2}\varepsilon w(M^*_H).$$

Moreover, $M^*_L := M^* \setminus M^*_H$ is an optimum solution to I'. We then compute a matching M' for I' using the Patching Lemma and output the feasible solution $M := M^*_H \cup M'$.

For a given choice of M^*_H the running time of the algorithm is dominated by the time to compute the two solutions M_1 and M_2. This can be accomplished in $O(m^{O(1)})$ time using Megiddo's parametric search technique. Hence the overall running time of the algorithm is $O(m^{p+O(1)})$, where the m^p factor is due to the guessing of M^*_H.

By our Patching Lemma, $w(M') \ge w(M^*_L) - 2w'_{\max}$. It follows that

$$\begin{aligned} w(M) = w(M^*_H) + w(M') &\ge w(M^*_H) + w(M^*_L) - 2w'_{\max} \\ &\ge w(M^*) - \varepsilon w(M^*_H) \ge (1-\varepsilon) w(M^*). \end{aligned}$$

□

3 Proof of the Patching Lemma

Let λ^* be the optimal Lagrangian multiplier and let M_1 and M_2 be two matchings of maximum Lagrangian weight $w_{\lambda^*}(M_1) = w_{\lambda^*}(M_2)$ such that $c(M_1) \le B \le c(M_2)$. Recall that M^* refers to an optimal solution to $(\bar{\Pi})$.

Observe that for $i \in \{1,2\}$ we have that

$$w_{\lambda^*}(M_i) + \lambda^* B \ge w_{\lambda^*}(M^*) + \lambda^* B \ge w_{\lambda^*}(M^*) + \lambda^* c(M^*) = \text{OPT}. \tag{1}$$

Also note that by the optimality of M_1 and M_2, $w_{\lambda^*}(e) \ge 0$ for all $e \in M_1 \cup M_2$.

We next show how to extract from $M_1 \cup M_2$ a matching M with the desired properties in polynomial time. As outlined above, our patching procedure proceeds in two phases:

Exchange phase: Consider the symmetric difference $M' = M_1 \oplus M_2$. Recall that $M' \subseteq M_1 \cup M_2$ consists of a disjoint union of paths $\mathcal{P}$ and cycles $\mathcal{C}$. We apply the following procedure until eventually $|\mathcal{P} \cup \mathcal{C}| \le 1$: Take some $X \in \mathcal{P} \cup \mathcal{C}$ and let $A := M_1 \oplus X$. If $c(A) \le B$ replace M_1 by A. Otherwise replace M_2 by A. Note that this way we maintain the invariant $c(M_1) \le B \le c(M_2)$.

Note that in each step the number of connected components in $M_1 \oplus M_2$ decreases; hence this procedure terminates after at most $O(n)$ steps. Moreover, by the optimality of M_1 and M_2 the Lagrangian weight of the two matchings does not change during the process, i.e., the two matchings remain optimal. To see this note that if there is some $X \in \mathcal{P} \cup \mathcal{C}$ such that $w_{\lambda^*}(M_1 \oplus X) < w_{\lambda^*}(M_1)$ then there must exist some $X' \in \mathcal{P} \cup \mathcal{C}$ such that $w_{\lambda^*}(M_1 \oplus X') > w_{\lambda^*}(M_1)$, which is a contradiction to the optimality of M_1. It follows that $w_{\lambda^*}(A) = w_{\lambda^*}(M_1) = w_{\lambda^*}(M_2)$.

Note that at the end of this phase we have for every $i \in \{1,2\}$

$$\begin{aligned} w(M_i) &= w_{\lambda^*}(M_i) + \lambda^* c(M_i) = w_{\lambda^*}(M_i) + \lambda^* B - \lambda^*(B - c(M_i)) \\ &\geq \mathrm{OPT} - \lambda^*(B - c(M_i)), \end{aligned} \tag{2}$$

where the inequality follows from (1).

In particular, if $c(M_i) = B$ for some $i \in \{1,2\}$, we are done: M_i is a feasible solution to the budgeted matching problem and $w(M_i) \geq \mathrm{OPT}$. Otherwise, we continue with the augmentation phase.

Augmentation Phase: The symmetric difference $M_1 \oplus M_2$ now consists of a unique path or cycle

$$X = (x_0, x_1, \ldots, x_{k-1}) \subseteq E$$

such that

$$c(M_1 \oplus X) = c(M_2) > B > c(M_1).$$

Observe that from (2) it follows that M_1 is a feasible solution whose original weight is close to optimal if its cost is sufficiently close to the budget B. The basic idea is to exchange edges along a subpath X' of X in order to obtain a feasible solution whose cost is close to the budget but still has large Lagrangian weight.

To this aim, we exploit the Gasoline Lemma. A formal statement is given below. We leave the proof of this lemma to the reader.

Lemma 2 (Gasoline Lemma). *Let $a_0, \ldots, a_{k-1}$ be a sequence of k real numbers such that $\sum_{j=0}^{k-1} a_j = 0$. There exists an index $i \in \{0, \ldots, k-1\}$ such that for every $h \in \{0, \ldots, k-1\}$,*

$$\sum_{j=i}^{i+h} a_{j \pmod{k}} \geq 0.$$

Consider the sequence

$$a_0 = \delta(x_0) w_{\lambda^*}(x_0), \quad a_1 = \delta(x_1) w_{\lambda^*}(x_1), \quad \ldots, \quad a_{k-1} = \delta(x_{k-1}) w_{\lambda^*}(x_{k-1}),$$

where $\delta(x_i) = 1$ if $x_i \in M_2$ and $\delta(x_i) = -1$ otherwise. (Note that, if X is a path, x_0 and x_{k-1} might both belong to either M_1 or M_2). This sequence has total value $\sum_{j=0}^{k-1} a_j = 0$ because of the optimality of M_1 and M_2. By the Gasoline Lemma there exists an edge x_i, $i \in \{0, 1, \ldots, k-1\}$, of X such that for any cyclic subsequence

$$X' = (x_i,\ x_{(i+1) \pmod{k}},\ \ldots,\ x_{(i+h) \pmod{k}}),$$

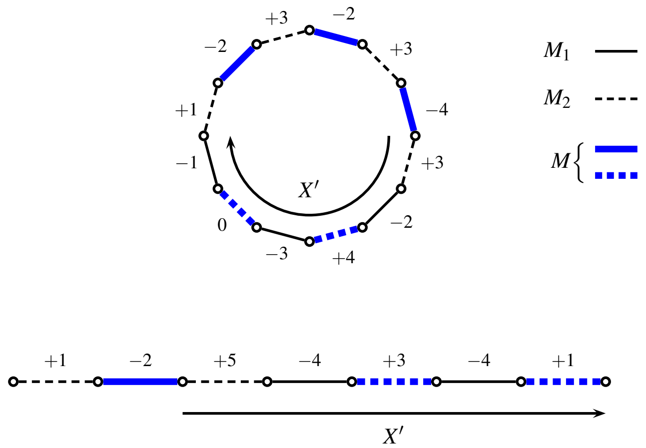

Fig. 2 Examples illustrating the construction used in the proof of the Patching Lemma. Each edge x_i is labeled with the value a_i.

where $h \in \{0,\ldots,k-1\}$, we have that

$$0 \leq \sum_{j=i}^{i+h} a_{j \pmod{k}} = \sum_{e \in X' \cap M_2} w_{\lambda^*}(e) - \sum_{e \in X' \cap M_1} w_{\lambda^*}(e). \tag{3}$$

Let X' be the longest such subsequence satisfying $c(M_1 \oplus X') \leq B$ (see Figure 2 for examples). Note that X' consists of either one or two alternating paths. (The latter case only occurs if X is a path whose first and last edge belong to X'). Let $e_1 = x_i$. Without loss of generality, we can assume $e_1 \in M_2$. (X' might start with one or two edges of M_1 with Lagrangian weight zero, in which case the next edge in M_2 is a feasible starting point of X' as well).

Observe that $M_1 \oplus X'$ is not a matching unless X is a path and e_1 its first edge. However, $M := (M_1 \oplus X') \setminus \{e_1\}$ is always a matching. Moreover,

$$c(M) = c(M_1 \oplus X') - c(e_1) \leq c(M_1 \oplus X') \leq B.$$

That is, M is a feasible solution to the budgeted matching problem.

It remains to lower bound the weight of M. We have

$$\begin{aligned} w(M_1 \oplus X') &= w_{\lambda^*}(M_1 \oplus X') + \lambda^* c(M_1 \oplus X') \\ &= w_{\lambda^*}(M_1 \oplus X') + \lambda^* B - \lambda^*(B - c(M_1 \oplus X')) \\ &\geq w_{\lambda^*}(M_1) + \lambda^* B - \lambda^*(B - c(M_1 \oplus X')) \\ &\geq \mathrm{OPT} - \lambda^*(B - c(M_1 \oplus X')), \end{aligned} \tag{4}$$

where the first inequality follows from (3) and the second inequality follows from (1).

Let $e_2 = x_{(i+h+1) \pmod k}$. The maximality of X' implies that $c(e_2) > B - c(M_1 \oplus X') \geq 0$. Moreover, by the optimality of M_1 and M_2, the Lagrangian weight of any edge $e \in M_1 \cup M_2$ is non-negative, and thus $0 \leq w_{\lambda^*}(e_2) = w(e_2) - \lambda^* c(e_2)$. Altogether

$$\lambda^*(B - c(M_1 \oplus X')) \leq \lambda^* c(e_2) \leq w(e_2)$$

and hence by (4)

$$w(M_1 \oplus X') \geq \mathrm{OPT} - w(e_2).$$

We conclude that

$$w(M) = w(M_1 \oplus X') - w(e_1) \geq \mathrm{OPT} - w(e_2) - w(e_1) \geq \mathrm{OPT} - 2w_{\max},$$

which proves the Patching Lemma. □

4 Extension and notes on the literature

The presented polynomial-time approximation scheme also extends to the *budgeted matroid intersection problem*. Here, we are given two matroids $\mathcal{M}_1 = (E, \mathcal{F}_1)$ and $\mathcal{M}_2 = (E, \mathcal{F}_2)$ on a common ground set of elements E. (We assume that these matroids are given implicitly by an *independence oracle*.) Moreover, we are given element weights $w : E \to \mathbb{Q}$, element costs $c : E \to \mathbb{Q}^+$ and a budget $B \in \mathbb{Q}^+$. The set of all feasible solutions $\mathcal{F} := \mathcal{F}_1 \cap \mathcal{F}_2$ is defined by the intersection of $\mathcal{M}_1$ and $\mathcal{M}_2$. The weight of an independent set $X \in \mathcal{F}$ is defined as $w(X) := \sum_{e \in X} w(e)$ and the cost of X is $c(X) := \sum_{e \in X} c(e)$. The goal is to compute a common independent set $X^* \in \mathcal{F}$ of maximum weight $w(X^*)$ among all feasible solutions $X \in \mathcal{F}$ satisfying $c(X) \leq B$.

Problems that can be formulated as the intersection of two matroids are, for example, matchings in bipartite graphs, arborescences in directed graphs, spanning forests in undirected graphs, etc. Although technically more involved, the ideas underlying our polynomial-time approximation scheme for the budgeted matching problem extend to this problem. More details can be found in [1].

Our algorithm needs to compute an optimal Lagrangian multiplier λ^* together with two respective optimal solutions. This can be done in polynomial time by standard techniques whenever the unbudgeted problem (Π) can be solved in polynomial time [9]. It can even be done in strongly polynomial time by using Megiddo's parametric search technique [5]. This technique can be used because *combinatorial* algorithms (only using comparisons and additions of weights) exist for (Π) (see, e.g., [10]). A similar idea was used by Goemans and Ravi [3] to derive strongly polynomial-time approximation scheme for the constrained minimum spanning tree problem.

The description of the Gasoline Puzzle was taken from the book "Combinatorial Problems and Exercises" by Lovász [4, Problem 3.21].

Naor et al. [7] proposed a fully polynomial-time approximation scheme (FPTAS) for a rather general class of problems, which contains the budgeted matching problem considered here as a special case. However, personal communication revealed that unfortunately the stated result [7, Theorem 2.2] is incorrect.

An interesting open problem is whether there is a fully polynomial-time approximation scheme for the budgeted matching problem. We conjecture that budgeted matching is not strongly NP-hard. However, finding an FPTAS for this problem might be a very difficult task because of its relation to the *exact perfect matching problem*: In this problem, we are given an undirected graph $G = (V,E)$ with edge weights $w : E \to \mathbb{Q}$ and a parameter $W \in \mathbb{Q}$. The goal is to find a perfect matching of weight exactly W (if it exists).

This problem was first posed in 1982 by Papadimitriou and Yannakakis [8]. The problem admits a polynomial-time Monte Carlo algorithm [2,6] if the edge weights are polynomially bounded. It is thus very unlikely that the exact perfect matching problem with polynomial weights is NP-hard because this would imply that RP = NP. However, the problem of finding a deterministic algorithm to solve the exact perfect matching problem remained open so far. For polynomial weights and costs the budgeted matching problem is equivalent to the exact perfect matching problem; see [1] for more details. As a consequence, a (deterministic) FPTAS for the budgeted matching problem would resolve a long-standing open problem.

References

1. A. Berger, V. Bonifaci, F. Grandoni, and G. Schäfer: Budgeted matching and budgeted matroid intersection via the gasoline puzzle. Mathematical Programming. 128:1–2, 355–372 (2011).
2. P. Camerini, G. Galbiati, and F. Maffioli: Random pseudo-polynomial algorithms for exact matroid problems. Journal of Algorithms. 13, 258–273 (1992).
3. R. Ravi and M. X. Goemans: The constrained minimum spanning tree problem. In: Karlsson, R., Lingas, A. (eds.), Proceedings of the 5th Scandinavian Workshop on Algorithms and Theory. Lecture Notes in Computer Science, vol. 1097, pp. 66–75. Springer, Berlin Heidelberg (1996).
4. L. Lovász: Combinatorial Problems and Exercises. North-Holland (1979).
5. N. Megiddo: Combinatorial optimization with rational objective functions. Mathematics of Operations Research. 4, 414–424 (1979).
6. K. Mulmuley, U. Vazirani, and V. Vazirani: Matching is as easy as matrix inversion. Combinatorica. 7, 105–113 (1987).
7. J. Naor, H. Shachnai, and T. Tamir: Real-time scheduling with a budget. Algorithmica. 47, 343–364 (2007).
8. C. H. Papadimitriou and M. Yannakakis: The complexity of restricted spanning tree problems. Journal of the ACM. 29, 285–309 (1982).
9. A. Schrijver: Theory of Linear and Integer Programming. Wiley (1986).
10. A. Schrijver: Combinatorial Optimization. Polyhedra and Efficiency. Springer, Berlin Heidelberg (2003).