The Clever Shopper Problem

Laurent Bulteau[1], Danny Hermelin[2], Anthony Labarre[1(✉)], and Stéphane Vialette[1]

[1] Université Paris-Est, LIGM (UMR 8049), CNRS, ENPC, ESIEE Paris, UPEM, 77454 Marne-la-Vallée, France
{laurent.bulteau,anthony.labarre,stephane.vialette}@u-pem.fr
[2] Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beersheba, Israel
hermelin@bgu.ac.il

Abstract. We investigate a variant of the so-called Internet Shopping problem introduced by Blazewicz et al. (2010), where a customer wants to buy a list of products at the lowest possible total cost from shops which offer discounts when purchases exceed a certain threshold. Although the problem is NP-hard, we provide exact algorithms for several cases, e.g. when each shop sells only two items, and an FPT algorithm for the number of items, or for the number of shops when all prices are equal. We complement each result with hardness proofs in order to draw a tight boundary between tractable and intractable cases. Finally, we give an approximation algorithm and hardness results for the problem of maximising the sum of discounts.

1 Introduction

Blazewicz et al. [3] introduced and described the Internet Shopping problem as follows: given a set of shops offering products at various prices and the delivery costs for each set of items bought from each shop, find where to buy each product from a shopping list at a minimum total cost. The problem is known to be NP-hard in the strong sense even when all products are free and all delivery costs are equal to one, and admits no polynomial $(c \ln n)$-approximation algorithm (for any $0 < c < 1$) unless P = NP.

A more realistic variant takes into account discounts offered by shops in some cases. These could be offered, for instance, when the shopper's purchases exceed a certain amount, or in the case of special promotions where buying several items together costs less than buying them separately. Blazewicz et al. [4] investigated such a variant, which features a concave increasing discount function on the products' prices. They showed that the problem is NP-complete in the strong sense even if each product appears in at most three shops and each shop sells exactly three products, as well as in the case where each product is available at three different prices and each shop has all products but sells exactly three of them at the same value. A variant where two separate discount functions

F. V. Fomin and V. V. Podolskii (Eds.): CSR 2018, LNCS 10846, pp. 53–64, 2018.
https://doi.org/10.1007/978-3-319-90530-3_6

are taken into account (one for the deliveries, the other for the prices) was also recently introduced and studied by Blazewicz et al. [5].

In this work, we investigate the case where a shopper aims to buy n books from m shops with free shipping; additionally, each shop offers a discount when purchases exceed a certain threshold (discounts and thresholds are specific to each shop). We show that the associated decision problem, which we call the CLEVER SHOPPER problem, is already NP-complete when only two shops are available, or when all books are available from two shops and each shop sells exactly three books. We also obtain parameterised hardness results: namely, that CLEVER SHOPPER is W[1]-hard when the parameter is m or the number of shops in a solution, and that it admits no polynomial-size kernel. On the positive side, we give a polynomial-time algorithm for the case where every shop sells at most two books, an XP algorithm for the case where few shops sell books at small prices, an FPT algorithm with parameter n, and another FPT algorithm with parameter m.

Let us now formally define CLEVER SHOPPER. For $n \in \mathbb{N}$, let $[n] = \{1, 2, \ldots, n\}$. Let B be a set of books to buy, S be a set of shops; $E \subseteq B \times S$ encodes the availability of the books in the shops, and $w : E \to \mathbb{N}$ encodes the prices. A subset $E' \subseteq E$ describes from which shop each book should be bought; each book is *covered exactly once* (i.e., any $b \in B$ has degree 1 in E'). A *discount* $d_s \in \mathbb{R}^+$ is associated to each shop s and offered when a *threshold* $t_s \in \mathbb{R}^+$ is reached, which is formally defined using the following *threshold function*:

$$\delta(s, E', d_s, t_s) = \begin{cases} d_s & \text{if } \sum_{(b,s)\in E'} w(e) \geq t_s, \\ 0 & \text{otherwise.} \end{cases}$$

We refer to the function $\mathscr{D}$ that maps each shop s to the pair (d_s, t_s) as the *discount function*. The problem we study is formally stated below, and generalises well-studied problems such as BIN COVERING [1] and H-INDEX MANIPULATION [12].

CLEVER SHOPPER

Input: an edge-weighted bipartite graph $G = (B \cup S, E, w)$; a discount function $\mathscr{D}$; a bound $K \in \mathbb{N}$.

Question: is there a subset $E' \subseteq E$ that covers each element of B exactly once and such that $\sum_{e\in E'} w(e) - \sum_{s\in S} \delta(s, E', d_s, t_s) \leq K$?

2 Hardness Results

We prove in this section several hardness results under various restrictions, both with regards to classical complexity theory and parameterised complexity theory. We show that CLEVER SHOPPER is NP-complete even if there are only two shops to choose from. For this first hardness result, we need book prices to be encoded in binary (i.e. they can be exponentially high compared to the input size).

Proposition 1. CLEVER SHOPPER *is* NP*-complete in the weak sense (*i.e., *prices are encoded in binary), even when* $|S| = 2$.

Proof (reduction from PARTITION*).* Recall the well-known NP-complete PARTITION problem [11]: given a finite set A and a size $\omega(a) \in \mathbb{N}$ for each element in A, decide whether there exists a subset $A' \subseteq A$ such that $\sum_{a\in A'} \omega(a) = \sum_{a\in A\setminus A'} \omega(a)$.

Let $\mathcal{I} = (A, \omega)$ be an instance of PARTITION, and $T = \sum_{a\in A} \omega(a)$. We obtain an instance $\mathcal{I}'$ of CLEVER SHOPPER as follows: introduce two shops s_1 and s_2 with $(d_{s_1}, t_{s_1}) = (d_{s_2}, t_{s_2}) = (1, T/2)$. Each item $a \in A$ is a book that shops s_1 and s_2 sell at the same price — namely, $\omega(a)$. It is now clear that there exists a subset $A' \subseteq A$ such that $\sum_{a\in A'} \omega(a) = \sum_{a\in A\setminus A'} \omega(a)$ if and only if all books can be purchased for a total cost of $T - 2$. □

This NP-hardness result allows arbitrarily high prices (the reduction from PARTITION requires prices of the order of $2^{|B|}$). In a more realistic setting, we might assume a polynomial bound on prices, i.e., they can be encoded in unary. As we show below, the problem remains hard for a few shops in the sense of W[1]-hardness. We complement this result with an XP algorithm in Proposition 7.

Proposition 2. CLEVER SHOPPER *is* W*[1]-hard for* $m = |S|$ *in the strong sense (*i.e.*, even when prices are encoded in unary).*

Proof (reduction from BIN PACKING*).* Recall the well-known BIN PACKING problem: given n items with weights $w_1, w_2, \ldots, w_n$ and m bins with the same given capacity W, decide whether each item can be assigned to a bin so that the total weight of the items in any bin does not exceed W. BIN PACKING is NP-complete in the strong sense and W[1]-hard for parameter m, even when $\sum_{i=1}^{n} w_i = mW$ and all weights are encoded in unary [10].

We build an instance $\mathcal{I}$ of CLEVER SHOPPER from an instance of BIN PACKING with the aforementioned restrictions as follows. Create m identical shops, each with $t_s = W$ and $d_s = 1$. Create n books, where book i is available in every shop at price w_i. The budget is $m(W - 1)$. In other words, any solution requires to obtain the discount from every shop, which is only possible if purchases amount to a total of exactly W per shop before discount. Therefore, the solutions to $\mathcal{I}$ correspond exactly to the solutions of the original instance of BIN PACKING. □

We can obtain another hardness result under the assumption that all books are sold at a unit price. Here we cannot bound the total number of shops (we give an FPT algorithm for parameter m in Proposition 8 in that setting), but only the number of *chosen* shops (i.e., shops where at least one book is purchased).

Proposition 3. CLEVER SHOPPER *with unit prices is* W*[1]-hard for the parameter* "number of chosen shops"*.*

Proof (reduction from PERFECT CODE*).* Given a graph $G = (V, E)$ and a positive integer k, PERFECT CODE asks for a size-k subset $V' \subseteq V$ such that for each vertex $u \in V$ there is precisely one vertex in $N[v] \cap V'$ (where $N[v]$ is the *closed* neighbourhood of v, i.e., v and its adjacent vertices, as opposed to the

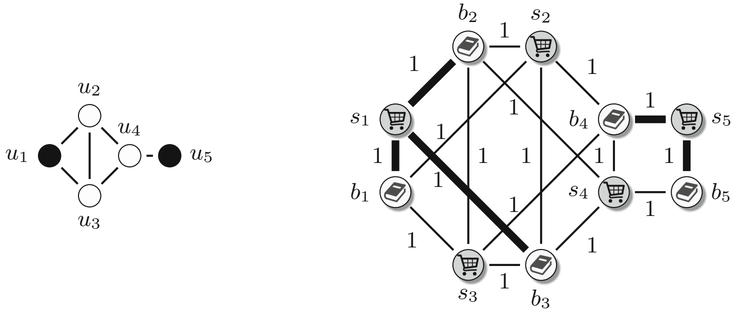

Fig. 1. Reducing PERFECT CODE to CLEVER SHOPPER. Left: the input graph with a size-2 perfect code (bold). Right: the corresponding bipartite graph and a solution with total cost $5 - 2 = 3$ (bold).

open neighbourhood $N(v) = N[v] \setminus \{v\}$). This problem is known to be W[1]-hard for parameter k [7].

Let $\mathcal{I} = (G = (V, E), k)$ be an instance of PERFECT CODE. Write $V = \{u_1, u_2, \ldots, u_n\}$. We obtain an instance $\mathcal{I}'$ of CLEVER SHOPPER as follows. Let us first define a bipartite graph $G' = (B \cup S, E')$ where $B = \{b_i : u_i \in V\}$, $S = \{s_i : u_i \in V\}$ and $E' = \{\{b_j, s_i\} : u_j \in N_G[u_i]\}$. All shops sell books at a unit price. As for the discount function, for each shop $s_i \in S$ we have $\mathscr{D}(s_i) = (1, d_G(u_i) + 1)$ (i.e., a unit discount will be applied, from $d_G(u_i) + 1$ of purchase). Figure 1 illustrates the construction.

We claim that there exists a size-k perfect code for G if and only if all books can be bought for a total cost of $n - k$.

$\boxed{\Rightarrow}$ Let $V' \subseteq V$ be a size-k perfect code in G. For every $u_i \in V$, let $u_{\mathrm{pc}(i)}$ be the unique vertex in $N[v] \cap V'$ (pc is well-defined since V' is a perfect code). Then buying each book $b_i \in B$ at shop $b_{\mathrm{pc}(i)}$ yields a solution for $\mathcal{I}'$, and it is simple to check that its cost is $n - k$.
$\boxed{\Leftarrow}$ Suppose that all books can be bought for a total cost of $n - k$. Since n books must be bought at unit price and shops only offer a unit discount, k shops must be chosen in the solution. Let $S' \subseteq S$ denote these k shops. Since $\mathscr{D}(s_i) = (1, d_G(u_i) + 1)$ for each shop $s_i \in S$, we conclude that for each book $b_i \in B$ there is precisely one shop in $N[b_i] \cap S'$. Then $\{u_i : s_i \in S'\}$ is a size-k perfect code in G.

Note that the number of visited shops corresponds exactly to the total discount received (i.e. to parameter k in the reduction). $\square$

We now prove[1] the non-existence of polynomial kernels (under standard complexity assumptions) for CLEVER SHOPPER parameterised by the number of books. To this end, we use the OR-COMPOSITION technique [6]: given a problem $\mathscr{P}$ and a parameterised problem $\mathscr{Q}$, an OR-COMPOSITION is a reduction taking t instances $(I_1, \ldots, I_t)$ of $\mathscr{P}$, and building an instance (J, k) of $\mathscr{Q}$, with k bounded

[1] Details will appear in the full version.

by a polynomial on $\max_{t' \leq t} |I_{t'}| + \log t$, such that (J, k) is a yes-instance if and only if there exists $t' \leq t$ such that $I_{t'}$ is a yes-instance. If $\mathscr{P}$ is NP-hard, then $\mathscr{Q}$ does not admit a polynomial kernel unless NP ⊆ coNP/poly [6].

Proposition 4. Clever Shopper *admits no polynomial kernel unless* NP ⊆ coNP/poly.

3 Positive Results

We now give exact algorithms for Clever Shopper: a polynomial-time algorithm for the case where every shop sells at most two books, and three parameterised algorithms based respectively on the number of books, the number of shops, and a bound on the prices.

We give a polynomial time algorithm for the case where each shop sells at most two books. As we shall see in Sect. 4, this bound is best possible. Its running time is dominated by the time required to find a maximum matching in a graph with $|B \cup S|$ vertices.

Proposition 5. Clever Shopper *is in* P *if every shop sells at most two books.*

Proof. Let $\mathcal{I}$ be an instance of Clever Shopper given by an edge-weighted bipartite graph $G = (B \cup S, E, w)$ and a pair (d_s, t_s) for each $s \in S$, where $d_s, t_s \in \mathbb{R}^+$. Vertices in S (resp. in B) have degree at most 2 (resp. at least 1). Note that vertices in S can be made to have degree exactly 2, by adding dummy edges with arbitrarily high costs, with no impact on the solution. For $b \in B$, let $p(b)$ be the cheapest available price for book b (discount excluded), i.e., $p(b) = \min\{w(\{b, s\}) \mid s \in S\}$.

Construct a new (non-bipartite) graph $G' = (B \cup S, E', w')$, as follows: for every shop $s \in S$, let $\{b_1, b_2\} = N_G(s)$ (i.e., the two books available at shop s).

- For each $i \in \{1, 2\}$, if $w(\{b_i, s\}) \geq t_s$, then add an edge $\{b_i, s\}$ to E' with weight $w'(\{b_i, s\}) = d_s + p(b_i) - w(\{b_i, s\})$.
- If $w(\{b_1, s\}) + w(\{b_2, s\}) \geq t_s$, add an edge $\{b_1, b_2\}$ to E' with weight $w'(\{b_1, b_2\}) = d_s + p(b_1) - w(\{b_1, s\}) + p(b_2) - w(\{b_2, s\})$. If edge $\{b_1, b_2\}$ existed already, keep only the one with maximum weight.

Note that edges with negative weights may remain: they may be safely ignored, but we keep them to avoid case distinctions in the rest of this proof. Figure 2 illustrates the construction. Since a maximum weight matching for G' can be found in polynomial time [8], it is now enough to prove the following claim: G' admits a matching of weight at least W if and only if instance $\mathcal{I}$ of Clever Shopper admits a solution of total cost at most $\sum_{b \in B} p(b) - W$.

$\boxed{\Leftarrow}$ Assume that instance $\mathcal{I}$ admits a solution $E^* \subseteq E$ of total cost $\sum_{b \in B} p(b) - W$. Note that $W \geq 0$ (the sum of the minimum prices of the books is an upper bound of the optimal solution). We build a matching M of G' as follows. Let $s \in S$ be any *discount shop*, i.e., a shop whose discount is claimed, and let b_1 and b_2 be its neighbours. Then at least one of them has to be bought from s to get the discount.

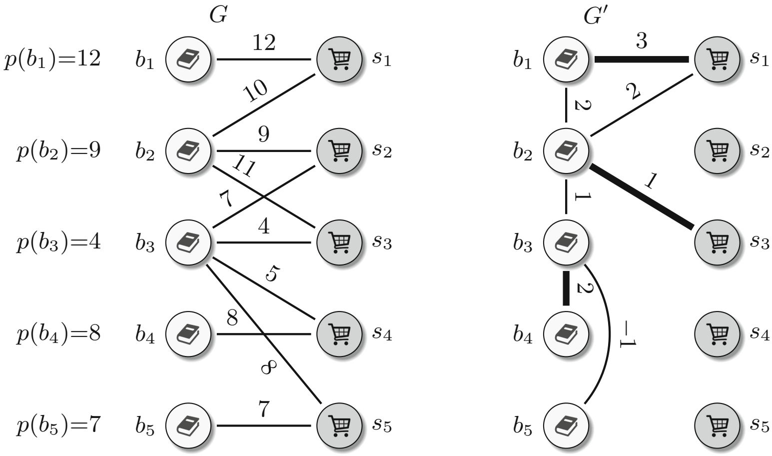

Fig. 2. Each shop offers a discount of 3 on a purchase of value ≥10. Bold edges indicate how to obtain optimal discounts: buy book b_1 from shop s_1, book b_2 from shop s_3, and books b_3 and b_4 from shop s_4. The remaining books are bought at their cheapest available price (so here we buy b_5 from s_5). Our clever customer used the discounts to buy all books for 6 less than if she had bought each book at its lowest price: 3 for b_1, 1 for b_2, 2 for b_3 and b_4 together.

- If $\{b_1, s\} \in E^*$ and $\{b_2, s\} \notin E^*$, add $\{b_1, s\}$ to M. The amount spent at this shop is $w(\{b_1, s\}) - d_s = p(b_1) - w'(\{b_1, s\})$.
- Similarly, if $\{b_2, s\} \in E^*$ and $\{b_1, s\} \notin E^*$, add $\{b_2, s\}$ to M. The amount spent at this shop is $w(\{b_2, s\}) - d_s = p(b_2) - w'(\{b_2, s\})$.
- Finally, if $\{b_1, s\} \in E^*$ and $\{b_2, s\} \in E^*$, then add $\{b_1, b_2\}$ to M. The amount spent at this shop is $w(\{b_1, s\}) + w(\{b_2, s\}) - d_s \geq p(b_1) + p(b_2) - w'(\{b_1, b_2\})$.

Note that edges added to M are indeed present in E', since in order to obtain the discount from s, the book prices must satisfy the same condition as for creating the corresponding edges. Note also that M is a matching, since each book can be bought from at most one shop. Let B^* be the set of books bought from discount shops. Summing over all these shops, the total price paid for the books in B^* is at least $\sum_{b \in B^*} p(b) - \sum_{e \in M} w'(e)$.

The books in $B \setminus B^*$ do not yield any discount, so the total price paid for them is at least $\sum_{b \in B \setminus B^*} p(b)$. Overall, the cost of the books is at least $\sum_{b \in B} p_b - \sum_{e \in M} w'(e)$, therefore $\sum_{e \in M} w'(e) \geq W$.

$\boxed{\Rightarrow}$ Let M be a maximum weight matching of G' of weight W. For each edge $e \in M$, let s_e be the shop for which e was introduced. For an edge $e = \{b, s_e\} \in M$, buy book b from shop s_e. The price is high enough to reach the threshold for the discount, so we pay $w(\{b, s_e\}) - d_e = p(b) - w'(e)$. For an edge $e = \{b_1, b_2\} \in M$, buy books b_1 and b_2 together from shop s_e. We again get the discount, and pay $w(\{b_1, s_e\}) + w(\{b_2, s_e\}) - d_e = p(b_1) + p(b_2) - w'(e)$. Note that for $e \neq f \in M$, $s_e \neq s_f$, so we never count the same discount twice. For every other book, buy them at the cheapest possible price $p(b)$, without expecting to get any discount. The total price paid is at most $\sum_{b\in B} p(b) - \sum_{e\in M} w'(e) = \sum_{b\in B} p(b) - W$. $\square$

We now give a dynamic programming FPT algorithm with the number of books as parameter.

Proposition 6. Clever Shopper *admits an* FPT *algorithm for parameter n with running time $O(m3^n)$.*

Proof. Given $j \in [m]$ and $B' \subseteq B$, let $p_j(B')$ be the price for buying all books in B' together from shop s_j (discount included), and $p_{\leq j}(B')$ be the lowest price that can be obtained when purchasing all books in B' from a subset of $\{s_1, \ldots, s_j\}$. Our goal is to compute $p_{\leq m}(B)$.

For $j = 1$, clearly $p_{\leq 1}(B') = p_1(B')$ for every B'. For any other j, consider an optimal way of buying the books in B' from shops $s_1, \ldots, s_j$. This way the customer buys some (possibly empty) subset B'' of books in s_j, and the rest, i.e., $B' \setminus B''$, at the lowest price from shops $s_1, \ldots, s_{j-1}$. Therefore:

$$p_{\leq j}(B') = \begin{cases} p_j(B') & \text{if } j = 1, \\ \min_{B''\subseteq B'}\{p_j(B'') + p_{\leq j-1}(B' \setminus B'')\} & \text{otherwise.} \end{cases}$$

The values of $p_j(B')$ for all j and B' can be computed in $O(m2^n)$ time. Then the dynamic programming table requires to enumerate, for all j, all subsets B' and B'' such that $B'' \subseteq B' \subseteq B$. Any such pair B'', B' can be interpreted as a vector $v \in \{0, 1, 2\}^n$, where $i \in B'' \Leftrightarrow v_i = 2$ and $i \in B' \Leftrightarrow v_i \geq 1$. Therefore, filling the dynamic table takes $m3^n$ steps, each requiring constant time. $\square$

As usual with dynamic programming, this algorithm yields the optimal price that can be obtained. One gets the actual solution (i.e., where to buy each book) with classic backtracing techniques.

The NP-hardness of Clever Shopper for two shops (using large prices, encoded in binary) and its W[1]-hardness when the parameter is the number of shops leave a very small opening for positive results: we can only consider small prices (encoded in unary) for a constant number of shops. The following result proves the tractability of this case.

Proposition 7. Clever Shopper *admits an* XP *algorithm running in time $O(nm\mathcal{W}^m)$, where $\mathcal{W}$ is the sum of all the prices of the instance, n is the number of books, and m is the number of shops.*

Proof. We propose the following dynamic programming algorithm, which generalises the classical pseudo-polynomial algorithm for Partition. Let $i \in [n]$ and $p_s \in [\mathcal{W}]$ for $s \in S$. Define $T[i, p_{s_1}, \ldots, p_{s_m}]$ as 1 if it is possible to buy books 1 to i by paying exactly p_s (discount excluded) in shop s; and 0 otherwise. For $i = 0$, $T[0, p_{s_1}, \ldots, p_{s_m}] = 1$ if and only if $p_s = 0$ for all $s \in S$. The following formula allows to fill the table recursively for $i \geq 1$:

$$T[i, p_{s_1}, \ldots, p_{s_m}] = \max_{e\in E, i\in e} T[i-1, p'_{s_1}, \ldots, p'_{s_m}] \text{ where } p'_s = \begin{cases} p_s - w(e) & \text{if } s \in e, \\ p_s & \text{otherwise.} \end{cases}$$

It remains to be checked whether the table contains a valid solution, which requires us to take the discounts into account. Clearly, an entry $T[n, p_{s_1}, \ldots, p_{s_m}] = 1$ leads to a solution if the following holds:

$$\sum_{s \in S} p_s - \sum_{s \in S, p_s \geq t_s} d_s \leq K.$$

The running time corresponds exactly to the time needed to fill the table: any of the $n\mathcal{W}^m$ cells requires at most m look-ups, which yields the claimed running time. □

Proposition 8. Clever Shopper *admits an* FPT *algorithm for parameter* m *when all prices are equal.*

Proof. We assume without loss of generality that all prices are equal to 1. Let $S' \subseteq S$. We write $f_{S'} : B \cup S \to \mathbb{N}$ for the following function:

$$f_{S'}(b) = 1 \text{ for } b \in B,$$
$$f_{S'}(s) = t_s \text{ for } s \in S',$$
$$f_{S'}(s) = 0 \text{ for } s \notin S'.$$

We write $d_{S'} = \sum_{s \in S'} d_s$ and $t_{S'} = \sum_{s \in S'} t_s$. An *$f$-star subgraph* of $G = (B \cup S, E)$ is a subgraph G' such that the degree of each vertex $u \in B \cup S$ is at most $f(u)$ in G', and every connected component of G' is isomorphic to $K_{1,p}$ for some integer p.

Let $\mathcal{I} = (B \cup S, E, w, \mathscr{D}, K)$ be an instance of Clever Shopper with $w(e) = 1$ for all $e \in E$. We show that $\mathcal{I}$ is a yes-instance if and only if there exists $S' \subseteq S$ with $|B| - d_{S'} \leq K$ such that $(B \cup S, E)$ admits an $f_{S'}$-star subgraph with $t_{S'}$ edges. An FPT algorithm follows easily from this characterisation: enumerate all subsets S' of S in time $2^{|S|}$, and for each subset, compute a maximum $f_{S'}$-star subgraph in time $O(|E| \log |B \cup S|)$ [9].

$\boxed{\Rightarrow}$ Let $E' \subseteq E$ be a solution and S' be the set of shops whose threshold t_s is reached. Since the total price is $|B| - d_{S'}$, we have $|B| - d_{S'} \leq K$. Since every weight equals 1, all vertices of S' have degree at most t_s in E'. Let $E'' \subseteq E'$ be a subset obtained by keeping exactly t_s edges incident to each $s \in S'$ and no edge incident to $s \notin S$. Then E'' is an $f_{S'}$-star subgraph of size $t_{S'}$.

$\boxed{\Leftarrow}$ Let $G' = (B \cup S, E')$ be an $f_{S'}$-star factor of G of size $t_{S'}$ with $S' \subseteq S$, and $|B| - d_{S'} \leq K$. The degree and size constraints force all vertices in S' to have degree exactly t_s in G'. We build a solution as follows: for each book $b \in B$, if E' contains an edge (b, s) incident to b, then buy b from shop s, otherwise buy b from any other shop. Overall, at least t_s books are purchased from a shop $s \in S'$, so the total price is at most $|B| - d_{S'}$. □

4 Approximations

Since variants of Clever Shopper are, by and large, hard to solve exactly, it is natural to look for approximation algorithms. However, our hardness proofs can be modified to imply the NP-hardness of deciding whether the total price (including discounts) is 0 or more. For instance, in Proposition 1, we can set the discounts to $T/2$ instead of 1, so the PARTITION instance reduces to checking whether the optimal solution has cost 0. Therefore, we start with the following bad news:

Corollary 1. CLEVER SHOPPER *admits no approximation unless* P = NP.

Since this result seems resilient to most natural restrictions on the input structure (bounded prices, bounded degree, etc.), our proposed angle is to maximise the total discount rather than minimise the total cost. However, maximising the total discount is only relevant when the base price of the books is the same in all solutions (otherwise the optimal solution might not be the one with maximum discount), i.e., each book b has a fixed price w_b, and $w(\{b, s\}) = w_b$ for every $\{b, s\} \in E$. We call this variant MAX-DISCOUNT CLEVER SHOPPER. This "fixed price" constraint is not strong (all reductions from Sect. 2 satisfy it). In this setting, Proposition 1 shows that it is NP-hard to decide whether the optimal discount is 1 or 2. This yields the following corollary:

Corollary 2. MAX-DISCOUNT CLEVER SHOPPER *is* APX*-hard: it does not admit a* $(2-\epsilon)$*-approximation unless* P = NP.

Whether or not MAX-DISCOUNT CLEVER SHOPPER admits a fixed-ratio approximation remains open.

Proposition 9. MAX-DISCOUNT CLEVER SHOPPER *is* APX*-hard even when each shop sells at most* 3 *books, and each book is available in at most 2 shops.*

Proof. We reduce from MAX 3-SAT (the problem of satisfying the maximum number of clauses in a 3-SAT instance), known to be APX-hard when each literal occurs exactly twice [2]. Let $\varphi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ be such a 3-CNF formula over a set $X = \{x_1, x_2, \ldots, x_n\}$ of boolean variables. For every $1 \leq i \leq m$ and $1 \leq j \leq 3$, let $\ell_{i,j}$ be the j-th literal of clause C_i. We obtain an instance $\mathcal{I}$ of MAX-DISCOUNT CLEVER SHOPPER by first building a bipartite graph $G = (B \cup S, E)$ as follows (for ease of presentation, C_i, x_i and $\ell_{i,j}$ will be used both to denote respectively clauses, variables and literals in 3-CNF formula context, and the corresponding vertices in G):

$$\begin{aligned}
B &= \{\ell_{i,j} : 1 \leq i \leq m \text{ and } 1 \leq j \leq 3\} \cup \{x_i : 1 \leq i \leq n\} \\
S &= \{C_i : 1 \leq i \leq m\} \cup \{t_i, f_i : 1 \leq i \leq n\} \\
E &= E_1 \cup E_{2,p} \cup E_{2,n} \cup E_3
\end{aligned}$$

where

$$\begin{aligned}
E_1 &= \{\{\ell_{i,j}, C_i\} : 1 \leq i \leq m \text{ and } 1 \leq j \leq 3\} \\
E_{2,p} &= \{\{\ell_{i,j}, t_i\} : 1 \leq i \leq m \text{ and } \ell_{i,j} \text{ is the positive literal } x_i\} \\
E_{2,n} &= \{\{\ell_{i,j}, f_i\} : 1 \leq i \leq m \text{ and } \ell_{i,j} \text{ is the negative literal } \overline{x_i}\} \\
E_3 &= \{\{x_i, t_i\}, \{x_i, f_i\} : 1 \leq i \leq n\}.
\end{aligned}$$

Observe that each shop sells exactly 3 books and that each book is sold in exactly 2 shops. We now turn to defining the prices, the thresholds and the discounts. All shops sell books at a unit price. For the shops C_i, $1 \leq i \leq m$, a purchase of value 1 yields a discount of 1. For the shops t_i and f_i, $1 \leq i \leq n$, a purchase of value 3 yields a discount of 2. This discount policy implies that, for every $1 \leq i \leq n$, a customer cannot obtain a 2 discount both in shop t_i and in shop f_i (this follows from the fact that the book x_i is sold by both shops t_i and f_i).

First, it is easy to see that the largest discount that can be obtained is $2n + m$ (the upper bound is achieved by obtaining a discount in every shop C_i for $1 \leq i \leq m$, and in either the shop t_i or the shop f_i for $1 \leq i \leq n$). On the other side, for any truth assignment τ for φ satisfying k clauses, a $2n + k$ discount can be obtained as follows.

- For any variable x_i, $1 \leq i \leq n$, if $\tau(x_i) = \texttt{false}$, then buy 3 books from shop t_i, and if $\tau(x_i) = \texttt{true}$ then buy 3 books from shop f_i. Intuitively, if a variable is true, then all negative literals are "removed" by f_i, and all positive literals remain available for the corresponding clauses.
- For any clause $C_i = \ell_{i,1} \vee \ell_{i,2} \vee \ell_{i,3}$ satisfied by the truth assignment τ, buy book $\ell_{i,j}$ from shop C_i, where $\ell_{i,j}$ is a literal satisfying the clause C_i.

Then it follows that

$$\begin{aligned}
\text{opt}(I) = 2n + \text{opt}(\varphi) &= 3m/2 + \text{opt}(\varphi) && \text{(since } 4n = 3m\text{)} \\
&\leq 3\ \text{opt}(\varphi) + \text{opt}(\varphi) && \text{(since } 2\ \text{opt}(\varphi) \geq m\text{)} \\
&\leq 4\ \text{opt}(\varphi).
\end{aligned}$$

Suppose now that we buy all books in B for a total discount of k'. First, we may clearly assume that $k' \geq 2n$ since a total $2n$ discount can always be achieved by buying 3 books either from shop t_i or from shop f_i, for every $1 \leq i \leq n$. Second, we may also assume that, for every $1 \leq i \leq n$, we buy either exactly 3 books from shop t_i or exactly 3 books from shop f_i. Indeed, if there exists an index $1 \leq i \leq n$ for which this is false, then buying either exactly 3 books from shop t_i or exactly 3 books from shop f_i instead results in a total k'' discount with $k'' \geq k'$ (this follows from the fact that we can get a 2 discount from t_i or f_i but only a 1 discount from any shop C_j, $1 \leq j \leq m$). We now obtain a truth assignment τ for φ as follows: for any variable x_i, $1 \leq i \leq n$, set $\tau(x_i) = \texttt{false}$ if we buy 3 books from shop t_i, and set $\tau(x_i) = \texttt{true}$ if we buy 3 books from shop f_i (the truth assignment τ is well-defined since, for $1 \leq i \leq n$, we cannot simultaneously buy 3 books from shop t_i and 3 books from shop f_i because of book x_i). Therefore, a clause C_i is satisfied by τ if and only if the corresponding shop C_i contains at least one book $l_{i,j}$ which is not bought from some other shop t_i or f_i. If we let k stand for the number of clauses satisfied by τ, then we obtain $k \geq k' - 2n$. It then follows that

$$\text{opt}(\varphi) - k = \text{opt}(I) - 2n - k \leq \text{opt}(I) - 2n - k' + 2n = \text{opt}(I) - k'.$$

Therefore, our reduction is an L-reduction (*i.e.*, $\text{opt}(\mathcal{I}) \leq \alpha_1 \text{opt}(\varphi)$ and $\text{opt}(\varphi) - k \leq \alpha_2 (\text{opt}(\mathcal{I}) - k'))$ with $\alpha_1 = 4$ and $\alpha_2 = 1$. □

Proposition 10. Max-Discount Clever Shopper *where each shop sells at most k books admits a k-approximation.*

Proof. Let B_s be the set of books sold by shop s. Our approximation algorithm proceeds as follows: start with a set of selected shops $S' = \emptyset$, a set of available books $B' = B$ and sort the shops by decreasing value of d_s. Then for each shop s, let $B'_s = B_s \cap B'$. If the books in B'_s are enough to get the discount ($\sum_{b \in B'_s} \geq t_s$), then assign all books of B'_s to shop s, add s to S' and set $B' = B' \setminus B'_s$. Finally, assign the remaining books to arbitrary shops that sell them.

We now prove the approximation ratio. For any $b \in B$, if $b \in B'_s$ for some $s \in S'$ then let $\delta(b) = d_s$, and $\delta(b) = 0$ otherwise. Thus, for any shop $s \in S'$, $d_s = \frac{1}{|B'_s|} \sum_{b \in B'_s} \delta(b) \geq \frac{1}{k} \sum_{b \in B'_s} \delta(b)$ due to the degree-k constraint. Note that for each shop of S', the amount spent at s is at least t_s, so the total discount obtained with this algorithm is $D \geq \sum_{s \in S'} d_s \geq \frac{1}{k} \sum_{b \in B} \delta(b)$.

We now compare the result of the algorithm with any optimal solution. For such a solution, let D^* be its total discount, S^* be the set of shops where purchases reach the threshold, and, for any $s \in S^*$, let B^*_s be the (non-empty) set of books purchased in shop s. Note that $D^* = \sum_{s \in S^*} d_s$.

Consider a shop $s \in S^*$. We show that there exists a book $b^*(s) \in B^*_s$ with $\delta(b^*(s)) \geq d_s$. If $s \in S^* \cap S'$, then we take $b^*(s)$ to be any book in B^*_s. Either $b^*(s) \in B'_s$, in which case $\delta(b^*(s)) = d_s$, or $b^*(s) \notin B'_s$, in which case $b^*(s)$ was assigned by the algorithm to a shop with a larger discount, i.e., $\delta(b^*(s)) \geq d_s$. If $s \in S^* \setminus S'$, since $s \notin S'$, at least one book in B^*_s is not available at the time the algorithm considers shop s; let $b^*(s)$ be such a book. Since it is not available, it has been selected as part of $B'_{s'}$ for some earlier shop s' (i.e., $d_s \leq d_{s'}$). Therefore, $b^*(s) \in B^*_s \cap B'_{s'}$ and $\delta(b^*(s)) = d_{s'} \geq d_s$. Since the sets B^*_s are pairwise disjoint for $s \in S^*$, we have $\sum_{s \in S^*} \delta(b^*(s)) \leq \sum_{b \in B} \delta(b)$. Putting it all together, we obtain:

$$D^* = \sum_{s \in S^*} d_s \leq \sum_{s \in S^*} \delta(b^*(s)) \leq \sum_{b \in B} \delta(b) \leq kD.$$

□

5 Conclusion

We introduced the Clever Shopper problem, a variant of Internet Shopping with free deliveries and shop-specific discounts based on shop-specific thresholds. We proved a number of hardness results, both in the classical complexity setting and from a parameterised complexity point of view. We also gave efficient algorithms for particular cases where restrictions apply to the number of books, the number of shops, or the nature of prices.

An interesting angle for future work is that of designing efficient exact algorithms for the general cases in which our FPT algorithms are not sufficient. Furthermore, it would be of interest to determine whether the Clever Shopper problem is FPT for parameter *maximum price + number of shops*.

References

1. Assmann, S., Johnson, D., Kleitman, D., Leung, J.-T.: On a dual version of the one-dimensional bin packing problem. J. Algorithms **5**, 502–525 (1984)
2. Berman, P., Karpinski, M., Scott, A.D.: Approximation hardness of short symmetric instances of MAX-3SAT. In: Electronic Colloquium on Computational Complexity (ECCC) (2003)
3. Blazewicz, J., Kovalyov, M.Y., Musial, J., Urbanski, A.P., Wojciechowski, A.: Internet shopping optimization problem. Appl. Math. Comput. Sci. **20**, 385–390 (2010)
4. Blazewicz, J., Bouvry, P., Kovalyov, M.Y., Musial, J.: Internet shopping with price sensitive discounts. 4OR **12**, 35–48 (2014)
5. Blazewicz, J., Cheriere, N., Dutot, P.-F., Musial, J., Trystram, D.: Novel dual discounting functions for the internet shopping optimization problem: new algorithms. J. Sched. **19**, 245–255 (2016)
6. Bodlaender, H.L., Jansen, B.M.P., Kratsch, S.: Kernelization lower bounds by cross-composition. SIAM J. Discret. Math. **28**, 277–305 (2014)
7. Cesati, M.: Perfect code is W[1]-complete. Inf. Process. Lett. **81**, 163–168 (2002)
8. Edmonds, J.: Paths, trees and flowers. Canad. J. Math. 449–467 (1965)
9. Gabow, H.N.: A note on degree-constrained star subgraphs of bipartite graphs. Inf. Process. Lett. **5**, 165–167 (1976)
10. Jansen, K., Kratsch, S., Marx, D., Schlotter, I.: Bin packing with fixed number of bins revisited. J. Comput. Syst. Sci. **79**, 39–49 (2013)
11. Karp, R.M.: Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W. (eds.) Proceedings of a Symposium on the Complexity of Computer Computations. The IBM Research Symposia Series, pp. 85–103. Plenum Press, Yorktown Heights (1972)
12. van Bevern, R., Komusiewicz, C., Niedermeier, R., Sorge, M., Walsh, T.: H-index manipulation by merging articles: models, theory, and experiments. Artif. Intell. **240**, 19–35 (2016)