Acta Mathematica Academiae Paedagogicae Nyíregyháziensis
28 (2012), 217–226
www.emis.de/journals
ISSN 1786-0091

THE DEBTS' CLEARING PROBLEM'S RELATION WITH COMPLEXITY CLASSES

CSABA PĂTCAŞ

Abstract. The debts' clearing problem is about clearing all the debts in a group of n entities (eg. persons, companies) using a minimal number of money transaction operations. In a previous paper we conjectured the problem to be NP-complete. In this paper we prove that it is NP-hard in the strong sense and also NP-easy. We also show the same results for a restricted version of the problem.

1. Introduction

In [4] we introduced the debts' clearing problem, and conjectured that it is NP-complete. In this paper we further investigate which complexity classes it belongs to based on the ideas of Benoist and Chauvet in proving similar results for Minimum Edge-Cost Flow in bipartite graphs ([1]).

The problem statement is the following:

Problem. *Let us consider a number of n entities (eg. persons, companies), and a list of m borrowings among these entities. A borrowing can be described by three parameters: the index of the borrower entity, the index of the lender entity and the amount of money that was lent. The task is to find a minimal list of money transactions that clears the debts formed among these n entities as a result of the m borrowings made.*

Example.

Borrower	Lender	Amount of money
1	2	3
2	3	2
3	4	5
4	5	6

2010 *Mathematics Subject Classification.* 68Q17, 05C21.
Key words and phrases. debt clearing, NP-hard, NP-easy, minimum edge-cost flow.

Solution.

Sender	Receiver	Amount of money
1	5	3
3	5	3
4	2	1

□

In [4] we modeled this problem using graph theory:

Definition 1. Let $G(V, A, W)$ be a directed, weighted multigraph without loops, $|V| = n$, $|A| = m$, $W : A \to \mathbb{Z}$, where V is the set of vertices, A is the set of arcs and W is the weight function. G represents the borrowings made, so we will call it the *borrowing graph*.

The borrowing graph corresponding to the example is shown in Figure 1.

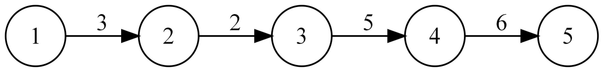

FIGURE 1. The borrowing graph associated with the given example. An arc from node i to node j with weight w means, that entity i must pay w amount of money to entity j.

Definition 2. Let us define for each vertex $v \in V$ the *absolute amount of debt* over the graph G:

$$D_G(v) = \sum_{\substack{v' \in V \\ (v,v') \in A}} W(v, v') - \sum_{\substack{v'' \in V \\ (v'',v) \in A}} W(v'', v).$$

Definition 3. Let $G'(V, A', W')$ be a directed, weighted multigraph without loops, with each arc (i, j) representing a transaction of $W'(i, j)$ amount of money from entity i to entity j. We will call this graph a *transaction graph*. These transactions clear the debts formed by the borrowings modeled by graph $G(V, A, W)$ if and only if: $D_G(v_i) = D_{G'}(v_i), \forall i = \overline{1, n}$, where $V = \{v_1, v_2, \ldots, v_n\}$

We will note this by: $G \sim G'$.

See Figure 2 for a transaction graph with minimal number of arcs corresponding to the example.

Using the terms defined above, the debt's clearing problem can be reformulated as follows:

Problem. *Given a borrowing graph $G(V, A, W)$ we are looking for a minimal transaction graph $G_{min}(V, A_{min}, W_{min})$, so that $G \sim G_{min}$ and $\forall G'(V, A', W') : G \sim G', |A_{min}| \leq |A'|$ holds.*

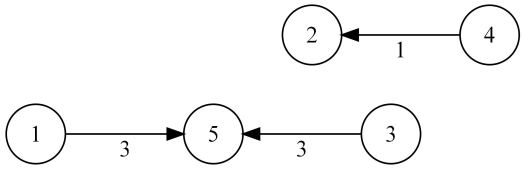

FIGURE 2. The respective minimum transaction graph. An arc from node i to node j with weight w means, that entity i pays w amount of money to entity j.

2. Related work

The problem was first discussed by Verhoeff in [5]. We were unaware of his paper while working on [4], so we use a slightly different terminology: what we defined above as the absolute amount of debt ($D_G(v_i)$) is called *balance* (b_i) in [5] and the transaction graph defined above is called *transfer graph.* Its definition is given by the *balancing relation*, which is similar to our "$\sim$" relation.

Verhoeff's paper concludes by noting the relationship with the SUBSET SUM and 3-PARTITION problems, saying that minimizing the number of transfers is at least as difficult as solving those problems. No full proofs are given regarding the complexity class of the problem.

In this work we give mathematically rigorous proofs for theorems stating, that the problem belongs to the NP-hard, strongly NP-hard, NP-easy and NP-equivalent classes. We also discuss a strongly restricted version, where the borrowing graph is only allowed to be a path.

3. The debts' clearing problem's relation with complexity classes

Let us note the optimization problem described in the introduction DEBT. We will call the corresponding decision problem DEBT-DECISION, defined as follows:

Problem. *Given a borrowing graph $G(V, A, W)$ and a natural number $M \leq |A|$, is there a transaction graph $G'(V, A', W'), G \sim G'$, so that $|A'| \leq M$?*

Lemma 4. DEBT-DECISION *is NP.*

Proof. It is easy to see, that the debts' clearing problem is NP: given a list of m transactions among n entities guessed by a nondeterministic algorithm, the D values (absolute amounts of debt) of the transaction graph can be easily computed in $\Theta(m)$ time, then compared to the original D values of the borrowing graph in $\Theta(n)$ time. □

Lemma 5. SUBSET SUM *is reducible to* DEBT-DECISION.

Proof. We will give a transformation from Karp's KNAPSACK problem ([3]), also called as SUBSET SUM([2])[1], showing that it is reducible to DEBT-DECISION.

The SUBSET SUM problem is defined as follows:

Problem. *Given a finite set S of positive integer numbers and a positive integer B, is there a subset $S' \subseteq S$, such that the sum of the elements in S' is exactly B?*

Let $S = \{s_1, s_2, \dots, s_n\}$. Let us construct a borrowing graph $G(V, A, W)$ in the following manner:

- $V = \{v_1, v_2, \dots, v_{n+2}\}$
- $A = \bigcup\limits_{i=1}^{k} \{(v_i, v_{n+1})\} \cup \bigcup\limits_{i=k}^{n} \{(v_i, v_{n+2})\}$, where k is chosen such that $\sum\limits_{i=1}^{k-1} s_i < B$ and $\sum\limits_{i=1}^{k} s_i \geq B$
- $W(v_i, v_{n+1}) = s_i, \forall i = \overline{1, k-1}$
- $W(v_k, v_{n+1}) = B - \sum\limits_{i=1}^{k-1} s_i$
- $W(v_k, v_{n+2}) = \sum\limits_{i=1}^{k} s_i - B$
- $W(v_i, v_{n+2}) = s_i, \forall i = \overline{k+1, n}$

We note that the above graph can be constructed in $\Theta(n)$ time. The resulting absolute amounts of debt will be:

- $D(v_i) = s_i, \forall i = \overline{1, n}$
- $D(v_{n+1}) = -B$
- $D(v_{n+2}) = B - \sum\limits_{i=1}^{n} s_i$

For example if we have $S = \{1, 2, 3, 6\}, B = 5$, the associated borrowing graph will be the one shown in Figure 3.

There is a transaction graph $G' \sim G$ with at most n arcs ($M = n$), if and only if there is a subset $S' \subseteq S$, such that the sum of the elements in S' is exactly B.

$\Rightarrow$: Let us suppose we have $G'(V, A', W') \sim G$ with $|A'| \leq n$. We must prove, that there is a subset S' satisfying the needed condition. From the signs of the D values it can be seen, that all arcs in A' should have one of the structures (v_i, v_{n+1}) or (v_i, v_{n+2}) with $i = \overline{1, n}$. This is because, if there would be at least one arc with a different structure, then there would be at least one $k \leq n$, so that v_k has no outgoing arc, thus $D_{G'}(v_k) \neq D_G(v_k)$, which contradicts $G \sim G'$.

But because $|A'| \leq n$, it means that each $v_i, i = \overline{1, n}$ is connected exclusively to v_{n+1} or v_{n+2}, but not both. Additionally $|A'|$ must be exactly n. Thus,

[1]The only difference between the two problems is, that KNAPSACK also allows negative integers

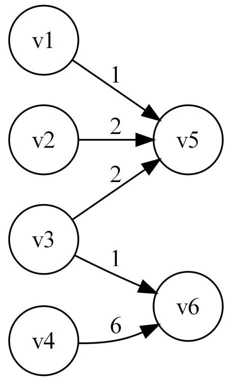

FIGURE 3. The borrowing graph associated with $S = \{1, 2, 3, 6\}, B = 5$ in the transformation from SUBSET SUM

the subset S' can be formed from the elements corresponding to the nodes connected to v_{n+1}.

$\Leftarrow$: Let us suppose that there is $S' \subseteq S$ with the sum of its elements being exactly B. We must prove, that there is a transaction graph $G'(V, A', W') \sim G$ with $|A'| \leq n$. From the D_G values it can be seen, that $|A'| \geq n$ (see Section 5.4. of [4] for a more detailed explanation). By constructing the transaction graph as below, we get exactly n arcs, the best solution possible:

- Let V' be the subset of nodes corresponding to S'.
- $A' = \bigcup_{v_i \in V'} \{(v_i, v_{n+1})\} \cup \bigcup_{v_i \in V \setminus V'} \{(v_i, v_{n+2})\}$
- $W'(v_i, v_{n+1}) = s_i$
- $W'(v_i, v_{n+2}) = s_i$ ☐

Theorem 6. DEBT-DECISION *is NP-complete.*

Proof. It follows directly from Lemma 4 and Lemma 5. ☐

Corollary 7. DEBT *is NP-hard.*

Proof. It follows directly from Theorem 6 (see also [2, page 114]). ☐

Lemma 8. 3-PARTITION *is pseudo-polynomially transformable in* DEBT-DECISION.

Proof. 3-PARTITION is defined as follows:

Given a finite set S of $3m$ positive integer elements, a positive integer bound B, such that each element from S is in the interval $(B/4, B/2)$ and the sum of the elements in S equals to $m \cdot B$, can S be partitioned in m disjoint sets, such that the sum of elements from each of them equals to B?

Let $S = \{s_1, s_2, \ldots, s_{3m}\}$. Let us construct the following $G(V, A, W)$ borrowing graph:

- $V = \{v_1, v_2, \ldots, v_{4m}\}$

- $A = \bigcup\limits_{i=1}^{m} \bigcup\limits_{j=k_i}^{k_{i+1}} \{(v_j, v_{3m+i})\}$, where
- $k_1 = 1, k_{m+1} = 3m$ and $k_2 \dots k_m$ are chosen such that none of the arc weights defined below results in a negative number, that is:

$$\sum_{i=k_{j-1}}^{k_j-1} W(v_i, v_{3m+j-1}) < B, \forall j = \overline{2, m}$$

and

$$\sum_{i=k_{j-1}}^{k_j} W(v_i, v_{3m+j-1}) \geq B, \forall j = \overline{2, m}$$

- $W(v_1, v_{3m+1}) = s_1$
- $W(v_i, v_{3m+j}) = s_i, \forall i = \overline{k_j + 1, k_{j+1} - 1}, \forall j = \overline{1, m}$
- $W(v_{k_j}, v_{3m+j-1}) = B - \sum\limits_{i=k_{j-1}}^{k_j-1} W(v_i, v_{3m+j-1}), \forall j = \overline{2, m}$
- $W(v_{k_j}, v_{3m+j}) = \sum\limits_{i=k_{j-1}}^{k_j} W(v_i, v_{3m+j-1}) - B, \forall j = \overline{2, m}$
- $W(v_{3m}, v_{4m}) = s_{3m}$.

This yields to $D(v_i) = s_i, \forall i = \overline{1, 3m}$ and $D(v_i) = -B, \forall i = \overline{3m+1, 4m}$. A transaction graph $G' \sim G$ will have at most $3m$ arcs ($M = 3m$) if and only if a partition of S into m disjoint subsets, all having the sum of elements equal to B exists.

$\Rightarrow$: Let $G'(V, A', W')$ be a transaction graph, with $|A'| \leq 3m$. From the D values it results by a similar argument to that from the proof of Lemma 5, that each arc starts at some $v_i, i = \overline{1, 3m}$ and goes to some $v_j, j = \overline{3m+1, 4m}$. Also $|A'| = 3m$ by the same argument, so there is exactly one such arc for each v_i. Thus, for each such v_j a corresponding subset can be constructed, that satisfies the needed condition.

$\Leftarrow$: Let $S_1, \dots, S_m$ be a partition of S, that is

$$\bigcup_{i=1}^{m} S_i = S,\ S_k \cap S_l = \emptyset,\ \forall k, l = \overline{1, m}, k \neq l,$$

and the sum of elements of S_i is exactly B for each $i = \overline{1, m}$. By associating each $s_i, \forall i = \overline{1, 3m}$ with v_i and $S_j, \forall j = \overline{1, m}$ with v_{3m+j}, then adding arcs from each element's associated node to the node associated with the subset that contains the element, we get a transaction graph with exactly $3m$ arcs. □

Theorem 9. Debt-decision *is NP-complete in the strong sense.*

Proof. By Lemma 4.1 from [2, page 101], Lemma 4 and Lemma 8 provide sufficient conditions for the proof of strongly NP-completeness. □

Corollary 10. Debt *is NP-hard in the strong sense.*

Proof. Follows directly from Theorem 9 (see also [2, page 115]). □

Let us define the problem Debt-decision-partial as follows:

Problem. *Given a borrowing graph $G(V, A, W)$, a "partial graph" $G^p(V, A^p, W^p)$ and a natural number $M \leq |A|$, can G^p "completed" to a transaction graph with at most M arcs? More formally is there a transaction graph $G'(V, A', W'), G \sim G'$, so that $|A'| \leq M$ and $A^p \subset A, W^p(a) = W'(a), \forall a \in A^p$?*

Lemma 11. Debt-decision-partial *is NP.*

Proof. The proof comes from the same logic as the proof of Lemma 4. □

Lemma 12. Debt *is Turing reducible to* Debt-decision-partial.

Proof. Let us note by $G^0(V, A^0, W^0)$, the graph having no arcs, that is $A^0 = \emptyset$. Assuming the existence of an algorithm $DDP(G, G^p, M)$, that solves Debt-decision-partial, Debt can be solved by the following algorithm:

Algorithm 3.1: Turing reduction of Debt to Debt-decision-partial

// Find the number of arcs in the optimal solution
$down := 0; up := |A|$;
while $down < up$ **do**
  $mid := \lfloor (down + up)/2 \rfloor$;
  if $DDP(G, G^0, mid)$ **then** $up := mid$;
  ;
  else $down := mid + 1$;
  ;
$MIN := down$;
$G' := G^0$;
// Find the arcs one by one
foreach $(u, v) \notin A' : D_G(u) > 0, D_G(v) < 0$ **do**
  $G^{aux} := G'$;
  $A^{aux} := A^{aux} \cup \{(u, v)\}$;
  $W^{aux}((u, v)) := \min\{|D_G(u) - D_{G'}(u)|, |D_G(v) - D_{G'}(v)|\}$;
  if $DDP(G, G^{aux}, MIN)$ **then** $G' := G^{aux}$;
  ;
  if $|A'| = MIN$ **then return**;
  G';

The above algorithm is based on the following ideas. The minimum transaction graph can contain no more than $|A|$ arcs (because $G \sim G'$), so the number of arcs (denoted by the variable MIN) can be found by a binary search. Then the solution can be built one arc at a time, trying out all the

$|\{u : D_G(u) > 0\}| \cdot |\{v : D_G(v) < 0\}|$ possibilities in the worst case. By choosing the weight to the value from line 11, we guarantee that at least one of the endpoints u and v will have its absolute amount of debt changed to zero after adding the arc. This condition is necessary to an optimal solution, because otherwise the addition of the arc would leave the number of "unsolved" nodes the same.

Since it is clear that DDP is called a polynomial number of times in the algorithm above, it means that we have a correct Turing reduction. □

Theorem 13. Debt *is NP-easy.*

Proof. By Theorem 6 Debt-decision is NP-complete, so any NP problem is (Turing-)reducible to it. By Lemma 11 Debt-decision-partial is NP, so it is reducible to Debt-decision. By Lemma 12 Debt is Turing-reducible to Debt-decision-partial, so by the transitivity of Turing-reducibility Debt is Turing-reducible to Debt-decision. From this and Lemma 4 the proof follows. □

Corollary 14. Debt *is NP-equivalent.*

Proof. Follows immediately from Corollary 7 and Theorem 13. □

4. A restricted version

Let us define the problem Debt-path as follows:

Problem. *Given a borrowing graph $G(V, A, W)$, whose arcs form a path, find the minimum transaction graph $G'(V, A', W'), G \sim G'$. More formally*

$$A = \bigcup_{i=1}^{n-1} \{(v_{p_i}, v_{p_{i+1}})\}, \quad v_{p_i} = v_{p_j} \Rightarrow i = j, \forall i, j = \overline{1, n}.$$

Theorem 15. Debt-path *is NP-hard.*

Proof. We can formulate Debt-path-decision in a similar way to Debt-decision, and prove that it's NP-complete by a reduction from Subset Sum. We build a path containing $n + 2$ nodes, and set:

$$W(v_i, v_{i+1}) = \sum_{j=1}^{i} s_j, \forall i = \overline{1, n}; \quad W(v_{n+1}, v_{n+2}) = B.$$

It is easy to see that the D values for this path will be similar to those from the proof of Lemma 5, the only difference being the swap between $D(v_{n+1})$ and $D(v_{n+2})$. □

For instance the path shown in Figure 4 can be also associated to $S = \{1, 2, 3, 6\}, B = 5$.

Theorem 16. Debt-path *is NP hard in the strong sense.*

Proof. We can reduce 3-Partition to Debt-path-decision as follows:

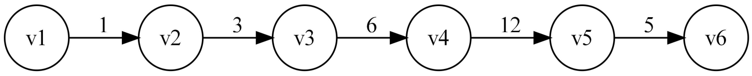

FIGURE 4. The path associated with $S = \{1, 2, 3, 6\}, B = 5$ in the transformation from SUBSET SUM

- $V = \{v_1, \ldots v_{4m}\}$
- $A = \bigcup_{i=1}^{4m-1} \{(v_i, v_{i+1})\}$
- $W(v_i, v_{i+1}) = \sum_{j=1}^{i} s_j, \forall i = \overline{1, 3m}$
- $W(v_i, v_{i+1}) = (4m - i) \cdot B, \forall i = \overline{3m + 1, 4m - 1}$.

This yields to exactly the same D values as in the proof of Lemma 8, thus the rest of the reasoning applies in this case too. □

Theorem 17. DEBT-PATH *is NP-easy.*

Proof. A constructive method similar to the one used in the proof of Theorem 13 is applicable. The details are left to the reader. □

Corollary 18. DEBT-PATH *is NP-equivalent.*

Proof. Readily follows from Theorem 15 and Theorem 17. □

5. CONCLUSIONS

We proved that the general optimization problem is NP-hard and also NP-hard in the strong sense. The latter result is important, because it follows that no pseudo-polynomial algorithm exists, that solves the problem (unless P = NP). Then we have shown, that the problem is also NP-easy, thus if P = NP it can be solved in polynomial time. From these results the NP-equivalency of the debts' clearing problem followed, which means that it can be solved in polynomial time if and only if P = NP.

In Section 4 we introduced a strongly restricted version, where the borrowing graph is allowed to be only a path and proved the same results for this version.

REFERENCES

[1] Thierry Benoist and Fabrice Chauvet. Complexity of some fpp related problems. Technical report, e-lab Research Report, 2001.

[2] Michael R. Garey and David S. Johnson. *Computers and intractability*. W. H. Freeman and Co., San Francisco, Calif., 1979. A guide to the theory of NP-completeness, A Series of Books in the Mathematical Sciences.

[3] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations (Proc. Sympos., IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972)*, pages 85–103. Plenum, New York, 1972.

[4] Csaba Pătcaş. On the debts' clearing problem. *Stud. Univ. Babeş-Bolyai Inform.*, 54(2):109–120, 2009.

[5] Tom Verhoeff. Settling multiple debts efficiently: An invitation to computing science. *Informatics in Education*, 3(1):105–126, 2004.

Department of Computer Science,
Babeş-Bolyai University,
M. Kogălniceanu 1, 400084 Cluj-Napoca, Romania
E-mail address: patcas.csaba@gmail.com