

- [7] V. G. Vizing. On an estimate of the chromatic class of a p-graph. *Diskret. Analiz.* **3** (1964), 25-30.
- [8] K. Wagner and G. Wechsung. *Computational Complexity*. D. Reidel Publishing Co (through Kluwer Academic Publishers, Holland), 1986.

$L = \Omega(n)$ . Therefore, at least for the case  $m = 3$ , the algorithm of Karzanov is faster and for the case  $m = 4$  not slower. In general, depending on the order of magnitude of  $L$  by  $n$ , the algorithm of Renegar has better complexity beginning from a certain value of  $m \geq 5$ .

**3.** The approach of this paper can be, in principle, easily generalized to the similar Knapsack problem, which is different from Safe Deposit Boxes in the more general goal function  $\sum_{i=1}^n c_i a_i$ . Then, the seniority is defined by “usefulness” per cost unit  $\frac{v_i}{c_i}$ , and Theorem 1, as well as its generalization to the case of general  $m$  remain valid. However, all this is of little use: the property “at most  $m - 1$  extra units” becomes an uninteresting upper bound  $m(c_{max} - 1)$  for the error, where  $c_{max} = \max_{i=1}^n c_i$ . (Observe that the solution obtained is approximate even for the case  $m = 1$ .)

**Acknowledgment.** The author is grateful to Bernard Moret for his moral support in reviving this paper.

## References

- [1] E. A. Dinic and A. V. Karzanov. Boolean Optimization Problems with Uniform Constraints. Reprint, Institute of Control Sci., Moscow, 1978, 42 p. (in Russian).
- [2] M. Fürer and B. Raghavachari. Approximating the minimum-degree Steiner tree to within one of optimal. *J. of Algorithms* **17** (1994), 409–423.
- [3] M. R. Garey and D. S. Johnson. *Computers and Intractability: a Guide to the Theory of NP-Completeness*. Freeman, San Francisco, 1979.
- [4] T. Jordan. On the Optimal Vertex Connectivity Augmentation, *J. of Combinatorial Theory, ser. B*, **63** (1995), 8–20.
- [5] B. M. E. Moret. *The Theory of Computation*. Addison-Wesley Publ. Co., Reading, Mass., 1997.
- [6] J. Renegar. A Polynomial Time Algorithm, Based on Newton’s Method, for Linear Programming. *Mathematical Programming* **40** (1988), 59-93.

at least  $n - m$  of those constraints correspond to variables; clearly, those variables are distinct. All those variables have integer values, i.e., at most  $m$  values out of  $\tilde{x}_i$  are not integer.

If no  $\tilde{x}_i$  is fractional,  $\{\tilde{x}_i\}$  is an optimum for the original Boolean Programming problem SDB, and we are done. Otherwise, let  $k$  be the number of  $\tilde{x}_i$  equal to 1. Then, the optimum of RSDB is strictly greater than  $k$  and, hence, the optimum of SDB is at least  $k + 1$ . Let us round every fractional  $\tilde{x}_i$  (i.e.,  $0 < \tilde{x}_i < 1$ ) to 0. Clearly, the obtained plan  $\{\lceil \tilde{x}_i \rceil\}$  is feasible for SDB. The goal function for  $\{\lceil \tilde{x}_i \rceil\}$ —the sum of variables—is at most  $k + m$ , i.e., it is an  $(m - 1)$ -approximation to the optimum for SDB, as required.

It can be easily shown that any optimum plan for the Linear Programming problem dual to RSDB forms an exchange rate such that the safe deposit boxes selected in the above approximation solution are the most valuable w.r.t. that rate, thus generalizing Theorem 1.

Karzanov, in [1], presents a brief description of a primal-dual algorithm to find an optimum for the linear relaxation as required; also a dual optimum plan is found. The complexity of this algorithm is  $O(mn + m^2 \cdot 2^{2m}(n + m)^m)$ . Observe that for any fixed  $m$  this complexity is  $O(n^{m+1})$ .

## 8 Other Algorithms and Discussion

**1.** Karzanov [1] improves the complexity for the case of two currencies to  $O(n^2)$ . He suggests another algorithm of the primal-dual type of geometrical nature (more complicated than the one suggested in this paper). To achieve the reduced complexity bound, he solves in  $O(n^2)$  time the following Computational Geometry problem: given  $n$  straight lines in the plane, sort, for each of them, its intersection points with the other straight lines w.r.t. their placement on it.

**2.** For the general case, let us turn to polynomial but not strongly polynomial algorithms. Let us consider a feasible instance of Linear Programming with  $n$  variables and  $O(n)$  constraints (as for RSDB, see Section 7), with integer data and the total bit length of the data  $L$ . By Renegar [6], one can find an exact solution to such a problem in  $O(n^3 L)$  approximate arithmetic operations, which totally takes  $O(n^3 L^2 \cdot \log(L) \cdot \log \log(L))$  bit operations. Observe that the order of magnitude of this bound by  $n$  is at least 5, since

exist? We reduce this problem to a series of  $n$  Deposit Safe Boxes problems, which proves NP-hardness of Deposit Safe Boxes.

Observe that, in order to solve a Partition problem, it suffices to solve  $n$  auxiliary problems (AP),  $k$ -th of which,  $k = 1, 2, \dots, n$ , differs from the original one in that the cardinality of the required subset is prefixed to be  $k$ . Let us model the  $k$ -th AP by the following instance of Safe Deposit Boxes. Let  $w_{max} = \max w_i$ , and let us set  $a_i = w_i, b_i = w_{max} - w_i, A = W, B = k \cdot w_{max} - W$ . For any feasible plan  $I$  holds

$$k \cdot w_{max} = A + B \leq \sum_{i \in I} a_i + \sum_{i \in I} b_i = \sum_{i \in I} (a_i + b_i) = |I| \cdot w_{max}.$$

Hence,  $|I| \geq k$ , and  $|I| = k$  iff  $\sum_{i \in I} a_i = A$ , i.e.,  $\sum_{i \in I} w_i = W$  (then  $\sum_{i \in I} b_i = B$ , too). It follows that if the  $k$ -th AP has a solution, then the optimum for the modeling problem equals  $k$  and the optimal solution is the required subset for the AP. On the other hand, if the  $k$ -th AP has no solution, then the optimum for the modeling problem exceeds  $k$ . Thus, to solve the  $k$ -th AP, it is sufficient to solve exactly the modeling instance of Safe Deposit Boxes.

*Remark:* the algorithm for approximate solution of Safe Deposit Boxes given in Section 3 together with the above reduction do not give rise to an approximate algorithm for Partition.

## 7 Case of More Than Two Currencies

In this Section, we show how to obtain an approximation with the error bounded by  $m - 1$  for Safe Deposit Boxes via its linear relaxation (this reduction is a joint work with A. V. Karzanov [1]).

Given an instance SDB of Safe Deposit Boxes for  $m$  currencies, let us consider it in the Boolean Programming form (see Introduction) and relax it to a Linear Programming problem RSDB by replacing the constraints  $x_i \in \{0, 1\}$  by the constraints  $x_i \geq 0$  and  $x_i \leq 1$ . Assume we are given an optimal solution  $\{\tilde{x}_i\}$  for RSDB which corresponds to an extreme point of the corresponding polyhedron in the  $n$ -dimensional space (such a solution always exists). Among the at least  $n$  constraints that are satisfied as equalities at this point, there are at most  $m$  corresponding to safe deposit boxes. Hence,

an event. Since there are  $\Theta(n^2)$  events, the complexity of this naive way is  $O(n^3 \log n)$ .

Now, we describe a Binary Search type algorithm based on the routine Check. Observe that, by the algorithm of Section 3, the following statement holds.

**Corollary 7** *For any non-degenerate instance of Safe Deposit Boxes, either one of the rates  $1 : 0$ ,  $0 : 1$  satisfies Theorem 1, or there exists an event such that, for any  $\beta$  inside the open interval between the events previous and next to it, the rate  $1 : \beta$  satisfies this Theorem.*

We suggest the following strategy for choosing rates for sequential calls of Check, till we succeed. The first iterations have rates  $1 : 0$ ,  $0 : 1$ , and  $1 : 1$ . Assume, w.l.o.g., that at the third iteration the dollar constraint is satisfied and the franc constraint is not. We double  $\beta$ :  $\beta = 2, 4, \dots$ , until the rate  $1 : 2^l$  gives feasibility by francs. Clearly, if a quasi-optimal solution is not found yet, then the right interval between events is placed inside the “current” interval  $2^{l-1}, 2^l$ . Then, we choose  $\beta$  to be the median  $\frac{3}{4}2^l$ . At any following iteration, we halve the current interval according to the constraint satisfied on the previous iteration, choose the median, and so on, making the value of  $\beta$  more and more precise. In the case where at the third iteration the franc constraint is satisfied, we similarly set  $\beta = \frac{1}{2}, \dots, 2^{-(l-1)}, 2^{-l}, \frac{3}{2}2^{-l}$ , and so on.

To bound the number of calls of Check, let us assume that all  $a_i, b_i$  are integers. Observe that the maximal event is bounded by  $a_{max} = \max_i a_i$  and the maximal denominator in the formula for an event is at most  $b_{max} = \max_i b_i$ . The latter fact implies that the minimal distance  $d$  between events is bounded from below by  $\frac{1}{b_{max}^2}$ . Therefore, after  $2 \log a_{max} + 2 \log b_{max} + const$  iterations, the consequent values of  $\beta$  will differ by less than  $d$ , which is sufficient for the current  $\beta$  to fall inside the right interval between events. Hence, the complexity of the suggested algorithm is  $O(n \cdot \log n \cdot \log(\max_i \{a_i, b_i\}))$ .

## 6 NP-Hardness

The following Partition problem is well known as NP-complete (see [3]): given a set of numbers  $w_i$ ,  $i = 1, 2, \dots, n$ , does a subset with the overall sum  $W$

can be rearranged in the reverse order, while the plan can be left without any changes. When this replacement is completed, the sequence of transpositions in the border group defined by the above lexicography can be undertaken and the plan changed accordingly.

The straightforward check shows that the sequence of replacements in the border group is as follows. First, the next to last by seniority box of the group is transposed with the last one, then the third from the end is transposed with the next to last and with the last one, becoming thereby the last box in the group, etc., till the most senior box having been consecutively transposed with all the others becomes the junior one. Based on this, it is easy to see that at any step of SAP the seniority order of boxes in the border group is as follows: first comes the subgroup consisting of a certain number of safe boxes having minimal values of  $a_i$ , in order of increase of  $a_i$ , then there are all the other boxes. The second subgroup is arranged in the decreasing order of  $a_i$ , with one possible exception that one box, immediately following by its value of  $a_i$  the boxes in the first subgroup, breaks the order as it is located not in the end of the subgroup but in some other place. Now, the validity of the Refinement readily follows from the fact that the plan  $I$  always consists of senior boxes.

## 5 Binary Search Algorithm

In this section, the non-degenerate case of Safe Deposit Boxes for two currencies is considered.

Observe that, given an exchange rate  $\alpha : \beta$ , the following routine **Check** either accepts it as satisfying Theorem 1 or rejects it and shows what is the direction to change it to the appropriate rate. The boxes are sorted according to the given exchange rate; the senior boxes are selected until one of the constraints is satisfied; if the other constraint is satisfied as well, or if it is satisfied after adding the senior non-selected box, then the rate is accepted; otherwise, the appropriate exchange rate must evaluate the currency corresponding to the unsatisfied constraint more. This routine works in  $O(n \log n)$  time.

Of course, it is sufficient to call Check for the rates  $1 : 0$  and  $0 : 1$  and, in case they do not provide a solution, for all the rates  $1 : \beta$ , where  $\beta$  is

4. The Transposition Array, according to the increase of values  $\beta'(i, j)$ , can be formed with the help of the above lexicographic ordering.

Thus, the combinatorial algorithm for the general case differs from the algorithm for non-degenerate problems only in somewhat more specified rule for defining the sequence of transposition array elements. It is not difficult to verify that the time complexity  $O(n^2 \log n)$  is not affected by such a complication.

Now we are able to prove the following **Refinement of Theorem 1** (an additional requirement for the quasi-optimal plan, the availability of which is assured by the Theorem): if the plan includes a part of a group of boxes of equal value, then this part consists of a certain number of boxes of the largest for this group dollar denomination and a certain number of boxes of the largest franc denomination.

In order to prove the Refinement, let us trace how the plan  $I$  is changing in the process of the algorithm. Let us divide the algorithm into parts corresponding to the fixed values of  $\beta$  (which are the events of the original problem). Let us select such an arbitrary part of the algorithm (further on called "SAP") and prove that within this part the plan always satisfies the condition of the Refinement. For short, we call "group" any group of boxes of identical value, in SAP. Using Pairwise Order Lemma, it is easy to see that at the beginning of SAP, boxes in each group are ordered by the decrease of dollar denomination, and at the end of SAP, the boxes are inversely ordered, i.e., by the decrease of franc denomination.

**Lemma 6** (i) *If at the beginning of SAP, a certain group completely belongs or completely does not belong to the plan, then the same property is preserved in the process of SAP;*

(ii) *If at the beginning of SAP, the plan border in the array of boxes ordered by seniority divided a certain group (the "border" group), then during the process of SAP the plan border either divides this group or immediately follows its last element.*

This Lemma can be proved by reduction ad absurdum, using Monotonicity Lemma.

Using the above Lemma, it is possible to amalgamate the SAP steps. At the very beginning of SAP, the boxes of every group, except the border one,

the transposition and the respective change of the plan results in feasibility by franc but not by dollar. Then, in order to find the quasi-optimal plan, it is obviously enough to try all the variants, namely, when the plan includes one, two, ..., all the boxes from the first group and the corresponding number of boxes from the second group.

Let us now turn to the general case. We begin with a theoretical elaboration and then describe an implementation.

It can be easily verified that an instance of Deposit Safe Box is not degenerate if all the numbers  $\beta(i, j) > 0$  are different and there are no identical boxes. Given an arbitrary safe deposit box problem, let us construct a non-degenerate problem equivalent to it. It is readily seen that addition of sufficiently small numbers  $\epsilon_i > 0$  to  $a_i$  keeps feasibility or infeasibility for all plans, and thus have no influence on the set of quasi-optimal solutions. W.l.o.g., let us assume that the numbering of the safe deposit boxes corresponds to the non-increasing order of the amount of dollars contained, i.e.,  $a_i \geq a_j \Leftrightarrow i \geq j$ . Let us choose a sequence of numbers  $\epsilon_i: 1 \gg \epsilon_1 \gg \epsilon_2 \gg \dots \gg \epsilon_n$  and set  $a'_i = a_i + \epsilon_i$ . For the modified problem, there are no identical boxes and the new events  $\beta'(i, j) = \beta(i, j) + \frac{\epsilon_i - \epsilon_j}{b_j - b_i} > \beta(i, j)$  are pairwise distinct as soon as the values for  $\epsilon_i, i = 1, 2, \dots, n$ , are pairwise sufficiently distinct; thus, the modified problem is not degenerate. It follows that, in order to solve the original problem, it is sufficient to construct a modified problem and to solve it using the above algorithm.

The proposed way of solution of a degenerate problem can be considerably simplified by making the algorithm combinatorial, as follows (among other things, there is no need to compute values for  $\epsilon_i$ ). The crucial observation is that the increasing ordering of  $\beta'(i, j)$  is defined by the increasing lexicographic order of triples  $(\beta(i, j), a_i, a_j)$ .

1. The considered numbering is defined by the box ordering at the initial phase of the algorithm.
2. Dealing with sums  $\sum'_a = \sum a'_i$ , the additional  $\epsilon_i$  can be not taken into consideration: using  $\sum_a$  instead of  $\sum'_a$  implies the same results.
3. There is no need to compute the new values for events  $\beta'(i, j)$ : instead of them, values  $\beta(i, j)$  can be used.

to  $\infty$  cannot cause a change of seniority ordering, since there are no events  $\beta(i, j) > \hat{\beta}$ , and from the fact that at  $\beta$  big enough, the ordering by  $v_i = 1 + \beta \cdot b_i$  coincides with the ordering by  $b_i$ . We add boxes following by seniority to our plan till the franc constraint is required. By Special Lower Bound Lemma, the resulting plan will be optimal.

This is the end of the description of the algorithm.  $\square$

The validity of the statement of Theorem 1 follows from the fact that at any point of the algorithm the plan includes boxes of maximum value w.r.t. the current exchange rate.

Let us bound the time complexity of this algorithm.

- The initial and final phases of the algorithm require  $O(n \log n)$  and  $O(n)$  time, respectively.
- For forming the Transposition Array, all the activity, excluding ordering the numbers  $\beta(i, j)$ , requires  $O(1)$  time per each pair  $(i, j)$  under consideration, i.e. total  $O(n^2)$  time. Sorting at most  $\frac{n(n-1)}{2}$  numbers  $\beta(i, j)$  requires  $O(n^2 \cdot \log n)$  time.
- Each iteration requires  $O(1)$  time, and we need at most  $\frac{n(n-1)}{2}$  iterations. Thus, the iterative part requires  $O(n^2)$  time.

It follows that the whole algorithm requires  $O(n^2 \cdot \log n)$  time (note that all the activity, excluding sorting the array of at most  $\frac{n(n-1)}{2}$  numbers, requires  $O(n^2)$  time).

## 4 Generalization for the Degenerate Case

Let us show how to adjust the above algorithm to the degenerate case.

First, let us consider the sub-case of degeneracy which arises as a result of existence of identical boxes (i.e.,  $a_i = a_j$ ,  $b_i = b_j$ ,  $i \neq j$ ), while under no exchange rate do three pairwise non-identical boxes have the same value. In this case, during an iteration of the algorithm, it can be required to transpose two neighboring groups of identical boxes in the seniority ordering. It is easy to see that the only non-trivial case is when: (i) the total set of boxes in both groups partially belongs and partially does not belong to the plan and (ii)

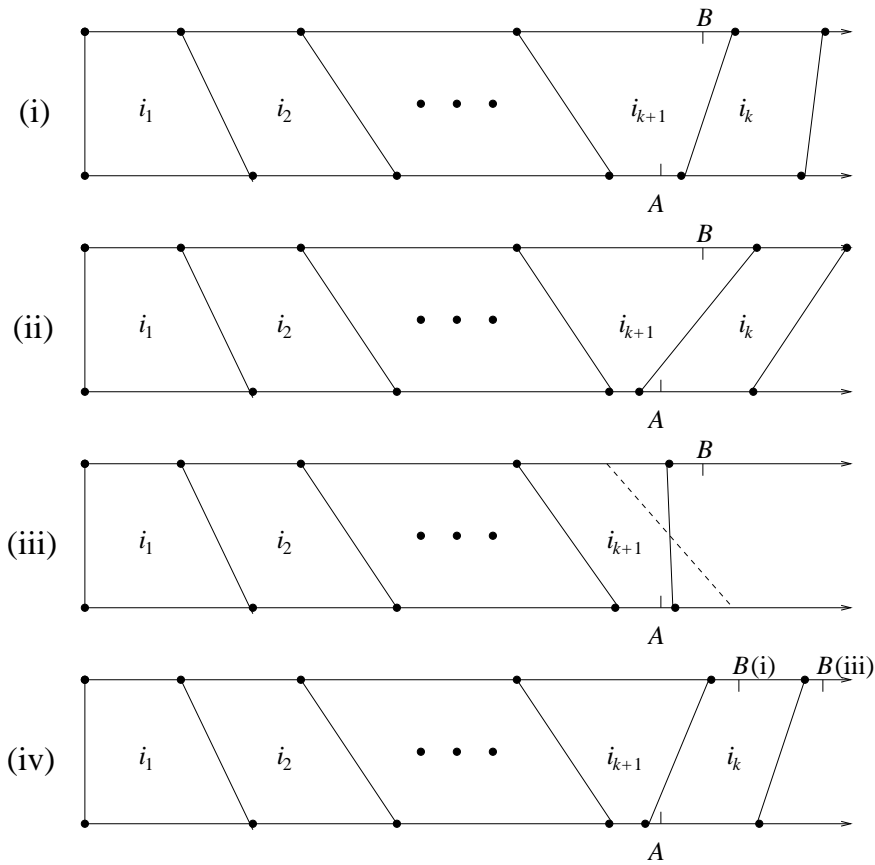


Figure 3: The four situations after a transposition.

(iv) The resulting plan  $I$  is infeasible under both constraints. Then, by Lower Bound Lemma,  $k + 1$  is a lower bound for the optimum. We return box  $i$  to the plan and thereby get the plan feasible by dollar (by Monotonicity Lemma). One of the situations (i) and (iii) arise, and we act as described for it.

If the current element of TA is the last one, this is the end of the iterative phase, otherwise, we proceed to the next iteration.

**Final Phase.**

Observe that the seniority ordering w.r.t. the current exchange rate  $1 : \tilde{\beta}$  is ordering by franc. This follows from the fact that increasing of  $\beta$  from  $\tilde{\beta}$

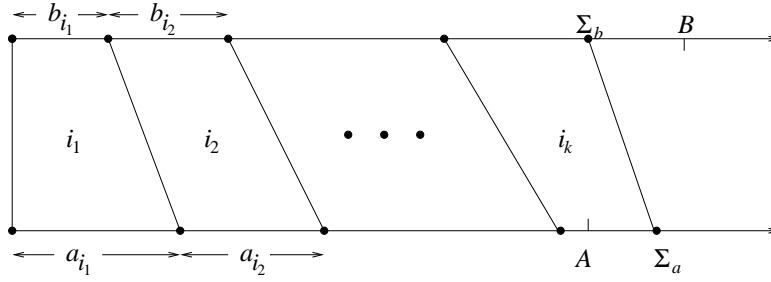


Figure 2: Initial phase without success.

such that the following conditions are fulfilled:  $\sum_a \geq A$ ,  $\sum_b < B$ , and if the least valuable box  $i_k$  is deleted from  $I$ , then  $\sum_a$  becomes strictly less than  $A$  (see Fig. 2). Note that, by Lower Bound Lemmas,  $k$  is a lower bound for the optimum.

**Iteration.** We set the new current element be the element of TA following the current one, and set  $\beta$  equal to the corresponding event. For each group at this element, we flip the seniority ordering for the two its boxes. Two cases are possible:

- $i, j \in I$  or  $i, j \notin I$ . Then the plan  $I$  (as non-ordered) is not changed.
- $i = i_k \in I, j = i_{k+1} \notin I$ . We exclude from  $I$  box  $i$  but include box  $j$ ; ( $\sum_a$  decreases, while  $\sum_b$  increases).

Evidently, there can occur at most one instance of the second case. If it occurs, the following four situations are possible (see Fig. 3).

- (i) The resulting plan  $I$  is feasible under both constraints. Then  $I$  is the optimal plan, since  $|I| = k$  is a lower bound for the optimum. The algorithm stops.
- (ii) The resulting plan  $I$  is feasible only by franc. Then we return box  $i$  to the plan. By Monotonicity Lemma, the resulting plan  $I$  becomes feasible under both constraints. Therefore,  $I$  is quasi-optimal, since  $|I| = k + 1$  and  $k$  is a lower bound for the optimum. The algorithm stops.
- (iii) The resulting plan  $I$  is feasible only by dollar (its franc deficiency is smaller than the previous one, since  $\sum_b$  has increased).

value. It is easy to see that boxes in each group are neighbors in both orders corresponding to the event intervals before and after  $\beta_0$ . When  $\beta$  passes from the interval before  $\beta_0$  to the interval after it, the ordering changes as follows: inside each group, the non-increasing order by franc value flips to the opposite order.

Let us call a safe deposit box problem **degenerate** if there exists an exchange rate  $\alpha : \beta$  with  $\alpha, \beta > 0$ , at which the values of some three boxes coincide. Note that a problem is not degenerate iff in the above picture no three lines intersect at the same point of the positive quadrant.

### 3 Combinatorial Algorithm

In this section, we present a combinatorial algorithm for the non-degenerate case. At any step of the algorithm, the values  $\alpha, \beta \geq 0$ , where  $\beta$  is an event, as well as a box ordering  $i_1, i_2, \dots, i_n$  at the event interval with the left end  $\beta$  will be defined; we will refer to it as “seniority order” at  $\beta$ . Given a plan  $I$ , let  $\sum_a$  denote  $\sum_{i \in I} a_i$  and  $\sum_b$  denote  $\sum_{i \in I} b_i$ . Let Transposition Array (TA) consist of the elements corresponding to the events in the increasing order, so that each element consists of the value  $\beta$  of the event and of the list of groups of boxes equal at  $\beta$ . Since we are in the non-degenerate case, each such group consists of exactly two boxes.

#### Algorithm.

**Initial Phase.** Let us set  $\alpha = 1$  and  $\beta = 0$  and order the boxes by seniority according to their dollar value and, when the dollar values are equal, according to their values in francs. Let us select the senior boxes till the plan  $I$  consisting of  $k$  senior boxes satisfies the dollar requirement,  $\sum_a \geq A$ ; note that, by Special Lower Bound Lemma,  $k$  is a lower bound for the optimum. If  $\sum_b \geq B$ , then, evidently,  $I$  is the optimal plan, and the algorithm stops.

If  $\sum_b < B$  (see an illustration in Fig. 2), we create Transposition Array, set the first element of TA to **current**, and proceed to Iterative Phase.

#### Iterative Phase.

**Inductive assumptions.** Before each iteration, the following holds. The value of  $\alpha$  is equal to 1. The value of  $\beta$  corresponds to the event at the current element of TA, and the order by seniority at  $\beta$  is given as an index sequence  $i_1, i_2, \dots, i_n$ . There is a plan  $I$  consisting of  $k$  senior boxes  $i_1, \dots, i_k$

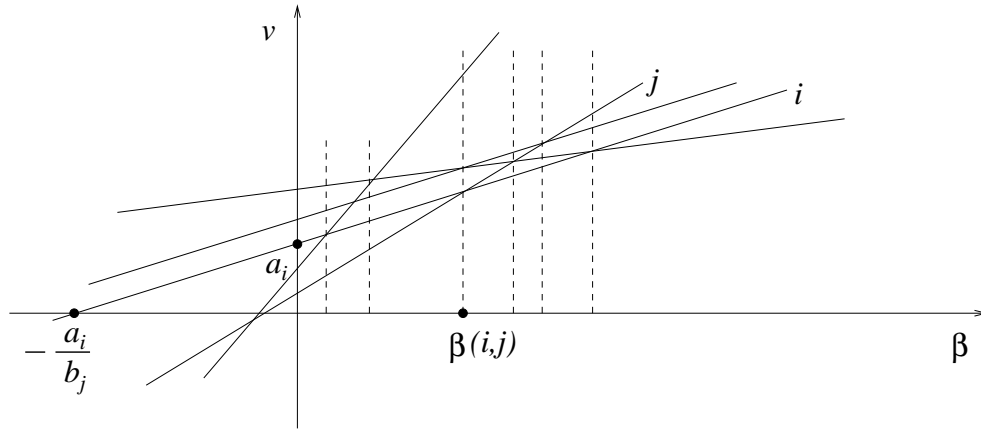


Figure 1: Geometry of events.

**Lemma 5 (Pairwise Order Lemma)** *Let  $\alpha = 1$ , and  $\beta$  vary in the range  $[0, \infty)$ . For any boxes  $i$  and  $j$  with  $b_i \leq b_j$ , the following holds:*

- (i) *if  $b_i < b_j$ , then: box  $i$  is more valuable than box  $j$  while  $\beta < \beta(i, j)$ ; they have the same value when  $\beta = \beta(i, j)$ ; box  $j$  is more valuable than box  $i$  as soon as  $\beta > \beta(i, j)$ ;*
- (ii) *if  $b_i = b_j$ , then the order connecting the values  $v_i$  and  $v_j$  is the same for all  $\beta$ .*

The validity of this Lemma is checked by direct comparison of values  $v_i = \alpha \cdot a_i + \beta \cdot b_i$  and  $v_j = \alpha \cdot a_j + \beta \cdot b_j$ .

Let us give a geometrical interpretation. In Figure 1, the straight lines represent graphically the dependencies  $v_i(\beta) = b_i\beta + a_i$ . The abscissa of the intersection point of  $i$ -th and  $j$ -th straight lines, if positive, is the event  $\beta(i, j)$ .

By Pairwise Order Lemma, all the events break the interval  $[0, \infty)$  for  $\beta$  into open intervals where the box ordering at  $\beta$  is unique and stable (uniqueness is up to transpositions of identical boxes, which will not be mentioned further). Moreover, the ordering on such an interval fits also its endpoints; thus the ordering is valid at the entire closed **event interval**.

In general, an event,  $\beta_0$ , can correspond to several pairs of boxes equally valuable at  $\beta_0$ ; the boxes in such pairs fall into groups with the same box

**Lemma 2 (Monotonicity Lemma)** *Let  $I$  and  $I'$  be two plans where  $I' \supset I$ . Then  $v(I') \geq v(I)$ . Moreover, if  $I$  is feasible w.r.t. some constraint, then  $I'$  is also feasible w.r.t. the same constraint.*

Optimality and quasi-optimality of plans constructed in the subsequent part of the paper are based on the following bounds.

**Lemma 3 (Lower Bound Lemma)** *Let there exist an exchange rate  $\alpha : \beta$  and an integer  $k$ , such that some plan consisting of  $k$  boxes most valuable w.r.t.  $\alpha : \beta$  is infeasible by both dollar and franc. Then the optimum is at least  $k + 1$ .*

**Proof:** Infeasibility of such a plan, say,  $I$ , by both dollar and franc implies that

$$v(I) = \sum_{i \in I} (\alpha \cdot a_i + \beta \cdot b_i) = \alpha \sum_{i \in I} a_i + \beta \sum_{i \in I} b_i < \alpha \cdot A + \beta \cdot B .$$

By the assumption of this Lemma and by Monotonicity Lemma,  $v(I)$  is maximal possible for the given exchange rate over all the plans consisting of  $k$  or a smaller number of boxes. Therefore, for any plan  $I'$ ,  $|I'| \leq k$ ,

$$v(I') \leq v(I) < \alpha \cdot A + \beta \cdot B$$

also holds. On the other hand, for any feasible plan  $I''$  we have

$$v(I'') = \alpha \sum_{i \in I''} a_i + \beta \sum_{i \in I''} b_i \geq \alpha \cdot A + \beta \cdot B .$$

Hence,  $|I''| \leq k$  holds for no feasible plan  $I''$ , as required.  $\square$

**Lemma 4 (Special Lower Bound Lemma)** *Let some plan infeasible by dollar (resp., by franc) consist of  $k$  boxes that are the most valuable by dollar (resp., by franc), i.e., w.r.t. the exchange rate  $1 : 0$  (resp.,  $0 : 1$ ). Then the optimum is at least  $k + 1$ .*

The proof is similar to that of Lemma 3.

Now, let us see how the ordering of boxes by value w.r.t. the rate  $1 : \beta$  (henceforth, **at**  $\beta$ ) changes when  $\beta$  changes from 0 to  $\infty$ . We call the value  $\beta(i, j) = \frac{a_i - a_j}{b_j - b_i}$ , if positive, an **event**. Observe that  $\beta(i, j) = \beta(j, i)$  for all  $i, j$ , and that  $b_i < b_j$  together with  $\beta(i, j) > 0$  imply  $a_i > a_j$ .

Besides, in Section 7, we refer to an algorithm (due to Karzanov [1]) solving Safe Deposit Boxes,  $m = 2$ , in time  $O(n^2)$  and discuss the use of polynomial Linear Programming algorithms to solve Safe Deposit Boxes for general  $m$ .

The first version of this paper was published in 1978 as a part of [1].

## 2 Duality Technique for the Case $m = 2$

In the following Sections, excluding Section 7, we consider the case of two currencies. A feasible solution is called **quasi-optimal** if the number of selected boxes is not greater than minimal possible plus one.

Let us introduce variables  $\alpha$  and  $\beta$ . We say that the **exchange rate**  $\alpha : \beta$  is defined if values  $\alpha \geq 0$  of one dollar and  $\beta \geq 0$  of one franc in some third (imaginary) currency are fixed. Let

$$v_i = v_i(\alpha, \beta) = \alpha \cdot a_i + \beta \cdot b_i$$

be the **value** of box  $i$  (w.r.t.  $\alpha : \beta$ ).

The validity of the following theorem stems from the algorithm given in the subsequent Sections.

**Theorem 1** *For any safe deposit box problem, there exists an exchange rate, such that a certain quasi-optimal plan consists of boxes most valuable according to this rate. Moreover, such a rate exists in the set  $\{1 : 0; 0 : 1; 1 : \frac{a_i - a_j}{b_j - b_i}, 1 \leq i, j \leq n\}$ .*

Here and in similar cases further, we mean a non-strict order: any selected box is at least as valuable as any non-selected box. In Section 4, a specification of this theorem for the case of equal values of several selected and non-selected boxes is presented. In Section 7, this theorem is generalized for the case of  $m > 2$ , and it is shown that the variables  $\alpha$  and  $\beta$  have sense of the dual variables for the integral relaxation of Safe Deposit Boxes.

A subset of boxes as well as the set of corresponding numbers  $I \in \overline{1, n}$  will be called a **plan**. Let us denote  $v(I) = \sum_{i \in I} v_i$ . A plan is called **feasible** w.r.t. the first constraint (by dollar) or w.r.t. the second constraint (by franc) if  $\sum_{i \in I} a_i \geq A$  or, respectively,  $\sum_{i \in I} b_i \geq B$ . Further on, we rely on the following simple statement stemming directly from non-negativity of  $a_i, b_i, \alpha$ , and  $\beta$ .

The latter type of problems is a kind of Knapsack, e.g., when it is necessary to fit in a maximal number of cargo units if there are upper bounds on the total weight and volume of the chosen cargo.

In analysis of problems of this class and in the presentation of methods for their solution we will use the safe deposit box model with  $m$  currencies (later referred to as “Safe Deposit Boxes”). Moreover, we assume that for any problem instance, all boxes together contain at least the required amount of each currency, i.e., that any instance is feasible.

It is clear how to solve any safe deposit box problem with one currency (say, dollars): the most valuable boxes should be selected till the required number of dollars is reached. This paper shows that there are analogies to the notion “box value” and to this way of solution for the case of several currencies.

The main results presented in this paper are as follows.

1. A combinatorial algorithm that solves Safe Deposit Boxes in the case of two currencies with the non-optimality at most 1 is presented. The algorithm time complexity is  $O(n^2 \cdot \log n)$  (see Sections 2–4).
2. Safe Deposit Boxes in the case of two currencies is proved to be NP-hard (see Section 6). Evidently, this implies the same result for any  $m$  greater than two.
3. For the general  $m$ , we observe (together with A. V. Karzanov) that for any instance of Safe Deposit Boxes, if an extreme-point (fractional) solution of its linear relaxation is given, the ceiling rounding of that solution is an approximation for the original problem with an absolute error of at most  $m - 1$ .<sup>2</sup> Karzanov [1] finds such a fractional solution with complexity  $O(n^{m+1})$ , for any  $m$  fixed. (See Section 7.)

Till now, in literature, there have been only a few non-trivial approximation algorithms with the absolute error bounded by a constant [7, 2, 4].

Furthermore, the technique developed gives rise to a Binary Search type algorithm. Such an algorithm for a non-degenerate two currency case with integer  $a_i, b_i$  is presented in Section 5; its complexity is  $O(n \cdot \log n \cdot \log(\max_i \{a_i, b_i\}))$ .

---

<sup>2</sup>Moret [5] calls solutions with the bounded absolute error “constant-distance approximations”.

# 1 Introduction

This paper deals with the following Boolean Programming problem:

$$\min \sum_{i=1}^n x_i ,$$

s.t.

$$\sum_{i=1}^n a_{ji}x_i \geq A_j, \quad j = 1, 2, \dots, m ;$$

$$x_i \in \{0, 1\}, \quad i = 1, 2, \dots, n ,$$

where  $a_{ji} \geq 0$  for all  $i, j$ . The main emphasis in this paper is on the case of two constraints ( $m = 2$ ). Let us consider two examples of such a problem.

- There are  $n$  safe deposit boxes, where box  $i$  contains  $a_i$  dollars and  $b_i$  francs. The problem is to open the minimal number of boxes so that the total amount of dollars obtained is at least  $A$  and of francs is at least  $B$ . (This original setting of the problem is due to A. C. Kronrod.<sup>1</sup>)

Of course, instead of dollars and francs there can be any non-convertible resources related to any objects (boxes). In particular, Moret [5] points out that in Operating Systems, breaking deadlocks by killing (or rolling back) the smallest number of processes to release resources needed for going on, can be formulated as such a problem.

- Let it be required to find two persons by phone, where the probabilities  $a_i$  and  $b_i$  of locating them using  $i$ th of the  $n$  given phone numbers are known. What is the minimal list of phone numbers assuring probabilities  $1 - \epsilon_1$  for locating the first person and  $1 - \epsilon_2$  for locating the second one?

The minimization problem with all constraints of the type  $\geq$  is equivalent to the corresponding maximization problem with all constraints of the type  $\leq$ . Switching from a problem of one type to the equivalent problem of the other is possible by replacement of the variables:  $y_i = 1 - x_i, i = 1, 2, \dots, n$ .

---

<sup>1</sup>To be more exact, Kronrod said “bank-note”; the term “safe deposit box” was used instead of it by Wagner and Wechsung in [8, Section 9.1.2] when citing the paper [1].

# Constant Absolute Error Approximation Algorithm for the “Safe Deposit Boxes” Problem

Yefim Dinitz \*

Dept. of Computer Science

Technion, Haifa, 32000, Israel

*e-mail:* dinitz@cs.technion.ac.il

July 8, 1997

## Abstract

Suppose there are  $n$  safe deposit boxes, each containing known amounts of  $m$  currencies, and there is a certain need for these currencies. The problem is how to open the minimal number of boxes in order to collect at least the prescribed amount of each currency. This problem is proved to be NP-hard. For the case  $m = 2$ , an algorithm with an absolute error of at most 1 is suggested, with complexity  $O(n^2 \log n)$ . For general  $m$ , we observe (together with A. V. Karzanov) that the ceiling rounding of an extreme-point solution for the corresponding integral linear program is an approximation with an absolute error of at most  $m - 1$ ; for  $m$  fixed,  $O(n^{m+1})$  time being sufficient to obtain such a solution.

---

\*Up to 1990, E. A. Dinic, Moscow.