Optimization Letters, , 1–10 (2009)

Knapsack Feasibility as an Absolute Value Equation Solvable by Successive Linear Programming

O. L. MANGASARIAN olvi@cs.wisc.edu
Computer Sciences Department
University of Wisconsin
Madison, WI 53706
Department of Mathematics
University of California at San Diego
La Jolla, CA 92093

Received September 2008; Revised September 2008

Editor: Panos Pardalos

Abstract. We formulate the NP-hard n-dimensional knapsack feasibility problem as an equivalent absolute value equation (AVE) in an n-dimensional noninteger real variable space and propose a finite succession of linear programs for solving the AVE. Exact solutions are obtained for 1880 out of 2000 randomly generated consecutive knapsack feasibility problems with dimensions between 500 and one million. For the 120 approximately solved problems the error consists of exactly one noninteger component with value in $(0,1)$, which when replaced by 0, results in a relative error of less than 0.04%. We also give a necessary and sufficient condition for the solvability of the knapsack feasibility problem in terms of minimizing a concave quadratic function on a polyhedral set. Average time for solving exactly a million-variable knapsack feasibility problem was less than 14 seconds on a 4 Gigabyte machine.

Keywords: knapsack feasibility, absolute value equations, successive linear programming

1. INTRODUCTION

We consider the n-dimensional NP-hard knapsack feasibility problem of finding an n-dimensional binary integer vector $y \in \{0,1\}^n$ such that:

$$a^T y = c, \tag{1}$$

where a is an n-dimensional column vector of positive integers, c is a positive integer and the superscript T denotes the transpose. We reformulate the problem as the following very simple absolute value equation (AVE) [11, 9, 10] in the n-dimensional real variable x:

$$\begin{array}{rcl} |x| & = & \frac{a}{2}, \\ e^T x & = & \frac{e^T a - 2c}{2}, \end{array} \tag{2}$$

where $|x|$ denotes the component-wise absolute value of x and e is a column vector of ones. As we shall establish in Section 2, a solution x of the AVE (2) yields immediately a solution y to the knapsack feasibility problem (1) and conversely as follows:

$$y_i = \frac{1}{2} - \frac{x_i}{a_i}, \;\; x_i = \frac{a_i}{2} - a_i y_i, \;\; i = 1, \ldots, n. \tag{3}$$

We shall propose the following quadratic program for solving the AVE (2) where x is represented as $x = r - s$:

$$\begin{array}{rcl} \min\limits_{(r,s)\in R^{2n}} & r^T s & \\ \text{s.t.} \ \ r+s & = & \frac{a}{2}, \\ e^T(r-s) & = & \frac{e^T a - 2c}{2}, \\ (r,s) & \geq & 0. \end{array} \tag{4}$$

We note immediately that the bilinear objective function of (4) will be zero at its global minimum solution (r,s) which yields an $x = r - s$ that will generate an exact solution y to the knapsack feasibility problem (1) through the (3). We also note that the variable s can be eliminated from the quadratic program (4) through the first constraint as follows:

$$s = \frac{a}{2} - r. \tag{5}$$

This results in the following simpler concave quadratic program in R^n instead of R^{2n}:

$$\begin{array}{rcl} \min\limits_{r\in R^n} & (a/2-r)^T r & \\ \text{s.t.} \ \ e^T r = & \frac{e^T a - c}{2}, & \\ r & \leq & \frac{a}{2}, \\ r & \geq & 0, \end{array} \tag{6}$$

the solution of which yields:

$$x = r - s = r - (\frac{a}{2} - r) = 2r - \frac{a}{2}. \tag{7}$$

Successive linearization of this simple concave quadratic programming problem terminates in a finite number of steps at a stationary vertex which, in our numerical experiments, yielded 1880 exact solutions for 2000 consecutive random knapsack feasibility problems attempted, and 120 all-integer approximate solutions with a relative error of less than 0.04%.

We now briefly describe the contents of the paper. In Section 2 we present the theory underlying the conversion of the knapsack feasibility problem to an absolute value equation and solving the latter via a concave quadratic minimization problem. In Section 3 we describe our algorithm for solving the knapsack feasibility problem by a finite succession of linear programs and in Section 4 we present our computational results. Section 5 concludes the paper.

A word about our notation now. All vectors will be column vectors unless transposed to a row vector by a superscript T. For a vector $x \in R^n$ the notation x_j will signify the j-th component. The scalar (inner) product of two vectors x and y in the n-dimensional real space R^n will be denoted by $x^T y$. For $x \in R^n$, $|x|$ denotes the vector in R^n whose components are the absolute values of the components of x. The notation $A \in R^{m\times n}$ will signify a real $m \times n$ matrix. For such a matrix, A^T will denote the transpose of A and A_i will denote the i-th row. A vector of ones in a real space of arbitrary dimension will be denoted by e. Thus for $e \in R^n$ and $x \in R^n$ the notation $e^T x$ will denote the sum of the components of x. A vector of zeros in a real space of arbitrary dimension will be denoted by 0. The abbreviation "s.t." stands for "subject to".

2. The Knapsack Feasibility Problem as a Simple Absolute Value Equation

It is well known that the n-dimensional knapsack feasibility problem is equivalent to the linear complementarity problem [1, 2, 5]

$$0 \leq z \perp Mz + q \geq 0, \tag{8}$$

where $M \in R^{k\times k}$, $q \in R^k$ and $\perp$ denotes orthogonality, that is $z^T(Mz+q) = 0$, and $k = n+2$ [1, 5] or $k = n+3$ [2]. It has also been shown in [11] that the linear complementarity problem is equivalent to the absolute value equation:

$$Ax - |x| = b, \tag{9}$$

where $A \in R^{k\times k}$ and $b \in R^k$. Deriving and simplifying the absolute value equation (9) that is equivalent to the knapsack feasibility problem (1), through the linear complementarity (8) equivalence, is somewhat lengthy and tedious. Instead we give now a simple direct proof of that equivalence.

Proposition 1 Knapsack Feasibility Equivalence to Absolute Value Equation *The knapsack feasibility problem (1) is equivalent to the absolute value equation (2) through the relations (3).*

Proof: Let y solve the knapsack feasibility problem (1). Define $x_i = \frac{a_i}{2} - a_i y_i$, $i = 1,\ldots,n$, as in (3). Then for $i = 1,\ldots,n$:

$$|x_i| = |\frac{a_i}{2} - a_i y_i| = \frac{a_i}{2}, \text{ if either } y_i = 0 \text{ or } y_i = 1. \tag{10}$$

Hence $|x| = \frac{a}{2}$. It follows from the definition of x above and $a^T y = c$ that:

$$e^T x = \frac{e^T a}{2} - c = \frac{e^T a - 2c}{2}. \tag{11}$$

Hence the AVE (2) is satisfied by x as defined in (3).

Conversely, let x solve the AVE (2). Define $y_i = \frac{1}{2} - \frac{x_i}{a_i}$, $i = 1,\ldots,n$, as in (3). Then we have that for $i = 1,\ldots,n$:

$$y_i = \frac{1}{2} - \frac{x_i}{a_i} = \left\langle \begin{array}{l} 0, \text{ if } x_i = \frac{a_i}{2} \\ 1, \text{ if } x_i = -\frac{a_i}{2} \end{array} \right\rangle, \tag{12}$$

where the second equality follows from (2) $|x| = \frac{a}{2}$. Thus, $y_i = 0$ or 1 for $i = 1,\ldots,n$. We also have from (12) above that:

$$a^T y = \sum_{i=1}^{i=n} a_i y_i = \sum_{i=1}^{i=n} \frac{a_i}{2} - \sum_{i=1}^{i=n} x_i = \frac{e^T a}{2} - e^T x = c, \tag{13}$$

where the last equality above follows from the second equation of (2). Hence $a^T y = c$ and y solves the knapsack feasibility problem (1). ■

We are ready now to relate a zero global minimum of the concave quadratic program (6) to that of the AVE (2) and consequently, by Proposition 1 above, to the knapsack feasibility problem (1).

Proposition 2 Knapsack Feasibility as a Concave Quadratic Program *The knapsack feasibility problem (1) has a solution y if and only the concave quadratic minimization problem (6) has a solution r such that:*

$$(\frac{a}{2}-r)^T r = 0. \tag{14}$$

Proof:

Necessity Let the knapsack feasibility problem have a solution y. The by Proposition 1 AVE (2) has a solution $x \in R^n$. Define r as:

$$r = (x)_+. \tag{15}$$

It follows then that

$$(x)_+ + (-x)_+ = |x| = \frac{a}{2}, \tag{16}$$

where the last equality from AVE (2). Hence

$$\frac{a}{2} - r = \frac{a}{2} - (x)_+ = (-x)_+ \geq 0, \tag{17}$$

where the second equality above follows from (16). Consequently the last two constraints of the quadratic program (6) are satisfied. Furthermore, by (15) and (17):

$$x = (x)_+ - (-x)_+ = r - \frac{a}{2} + r = 2r - \frac{a}{2}, \tag{18}$$

and hence by AVE (2):

$$\frac{e^T a - 2c}{2} = e^T x = 2e^T r - \frac{e^T a}{2}, \tag{19}$$

or equivalently,

$$2e^T r = \frac{2e^T a - 2c}{2}. \tag{20}$$

Thus the first constraint of the quadratic program (6) is satisfied. Finally the nonnegative objective function of the quadratic program (6) satisfies:

$$(\frac{a}{2} - r)^T r = (-x)_+^T (x)_+ = 0, \tag{21}$$

where x solves the AVE (2). Hence the quadratic program (6) is solvable by r as defined above and has a global minimum of zero.

Sufficiency At a zero minimum of the quadratic program (6) define

$$x = 2r - \frac{a}{2} = r - (\frac{a}{2} - r). \tag{22}$$

Then since $r^T(\frac{a}{2} - r) = 0$, $r \geq 0$ and $\frac{a}{2} - r \geq 0$, it follows that

$$|x| = r + \frac{a}{2} - r = \frac{a}{2}, \tag{23}$$

which is the first equation of AVE (2). Furthermore,

$$e^T x = 2e^T r - \frac{e^T a}{2} = \frac{2e^T a - 2c}{2} - \frac{e^T a}{2} = \frac{e^T a - 2c}{2}, \tag{24}$$

where the second equality follows from the first constraint of the quadratic program (6). Hence, x solves AVE (2) and by Proposition 1 the knapsack feasibility problem is solvable by $y_i = \frac{1}{2} - \frac{x_i}{a_i}$, $i = 1, \ldots, n$. ■

Based on the results above we are ready now to specify our algorithm for solving the knapsack feasibility problem (1).

3. Solving the Knapsack Feasibility Problem by Successive Linear Programming

We shall solve the concave quadratic programming minimization problem (6) by the following finitely terminating successive linearization algorithm.

Algorithm 1 SLA: Successive Linearization Algorithm for the Quadratic Program (6)
Start with a random $r^0 \in R^n$. Given $r^i \in R^n$ compute r^{i+1} by solving the following linearization of the quadratic program (6):

$$\begin{array}{rl} \min\limits_{r \in R^n} & (\frac{a}{2} - 2r^i)^T r \\ s.t. & e^T r = \frac{e^T a - c}{2}, \\ & r \leq \frac{a}{2}, \\ & r \geq 0. \end{array} \tag{25}$$

Stop when:

$$(\frac{a}{2} - 2r^i)^T (r^{i+1} - r^i) = 0. \tag{26}$$

By making use of [6, Theorem 4.2] we have the following result.

Proposition 2 Finite Termination of SLA *The successive linearization algorithm 1 generates a finite sequence of iterates $\{r^1, r^2, \ldots, r^{\bar{i}}\}$ with strictly decreasing objective function values $\{(\frac{a}{2} - r^i)^T r^i\}$ of the quadratic program (6) objective function terminating at $r^{\bar{i}}$ that satisfies the following minimum principle necessary optimality condition [4, Theorem 5.2.4]:*

$$(\frac{a}{2} - 2r^{\bar{i}})^T (r - r^{\bar{i}}) \geq 0, \ \forall \text{ feasible } r \text{ of } (6). \tag{27}$$

We note immediately that even though the minimum principle necessary optimality condition (27) is not a sufficient optimality condition for the quadratic program (6), points satisfying (27) have been very effective in solving nonconvex optimization problems such as those in [8, 7, 6] as well as our quadratic program (6), which will be demonstrated by the computational results of the next section. Consequently, we now propose the following algorithm for solving the knapsack feasibility problem (1) that is based on Propositions 1, 2 and 2.

Algorithm 3 SLA: Successive Linear Program for the Knapsack Feasibility Problem (1)

(I) *Apply Algorithm 1 to the quadratic program (6) until termination at $r^{\bar{i}}$ and, by (7), set $x = 2r^{\bar{i}} - \frac{a}{2}$. If ${r^{\bar{i}}}^T(a - r^{\bar{i}}) = 0$ then, by (3), $y_i = \frac{1}{2} - \frac{x_i}{a_i}$, $i = 1, \ldots, n$ solves the knapsack feasibility problem (1). Stop.*

(II) *If ${r_\ell}^T(a - r_\ell) > 0$ for some ℓ, compute $x_\ell = 2r_\ell - \frac{a}{2}$ by (7), $y_\ell = \frac{1}{2} - x_\ell/a_\ell$ by (3) and $v_\ell = a_\ell y_\ell$.*

(III) *Set $y_\ell = 0$ and find an a_i such that $a_i = v_\ell$ and $y_i = 0$. If no such a_i exists, the current y solves the knapsack feasibility only approximately. Stop.*

(IV) *Set $y_i = 1$. The vector y so modified solves the knapsack feasibility problem exactly. Stop.*

Remark 4 *Note that in the above algorithm we assume that at worst for only one component ℓ is $r_\ell(a_\ell - r_\ell) > 0$. This assumption as well as the integrality of v_ℓ is validated empirically by every single solution obtained by the SLA Algorithm 1 for a solvable knapsack feasibility problem.*

Remark 5 *We note that a concave quadratic program similar to (6) can be generated directly for the knapsack feasibility problem (1) without transforming it into an AVE as follows:*

$$\begin{array}{rrcl} \min\limits_{y \in R^n} & e^T(e - y) & & \\ s.t. & a^T y & = & c, \\ & y & \leq & e, \\ & y & \geq & 0. \end{array} \tag{28}$$

A solution to this problem obtained by a successive linearization algorithm similar to Algorithm 1 also leads to a solution y with one noninteger component and $n-1$ integer components because at least $n-1$ components of the the constraints $0 \leq y \leq e$ are satisfied as equalities at a vertex solution of the linearization of the concave quadratic program (28). However, utilizing this quadratic programming formulation instead of the AVE-based quadratic (6), Algorithm 3 performs very poorly by comparison. For example, testing it on 100 random consecutive knapsack feasibility problems of dimension $n = 10,000$, this approach did not solve a single problem exactly. In contrast, utilizing the AVE-based quadratic program (6), Algorithm 3 solved 99 out of these 100 problems exactly. We have no theoretical justification for this poor performance resulting from utilizing the concave quadratic program (28), instead of the quadratic program (6). One possibility is that the program (28) obtains such a unique vertex, that it makes it nearly impossible to implement the first part of Step III of Algorithm 3 in order to generate an exact solution from an approximate solution that has only one noninteger component. Another possibility is that the formulation (28) seeks an integer solution y directly, whereas our formulation (6) seeks a noninteger solution r which might make its task easier.

We turn now to numerical testing of Algorithm 3.

4. Computational Results

We tested our AVE formulation (2) by solving two sets of 1000 random knapsack feasibility problems (1). Each problem of the first set was generated and formulated as the concave quadratic program (6) as follows:

1. Each component of the vector a of (1) was a positive integer picked from a uniform distribution over the interval $[0,100]$.

2. A solution y to (1) was picked as a random n-dimensional binary integer vector from $\{0,1\}^n$, with approximately one half of its components being zeros and the other half being ones. The integer c was then computed as $c = a^T y$.

3. One thousand random knapsack feasibility problems were generated *consecutively* in groups of 100 for each of the following values of n and solved by the quadratic program (6):

$$n = 500, 1000, 5000, 10000, 20000, 50000, 100000, 200000, 500000, 1000000. \quad (29)$$

4. Of these 1000 consecutive randomly generated knapsack feasibility problems, Algorithm 3 solved 990 problems exactly while 10 were solved approximately as detailed in the next comment.

5. For each of the 10 approximately-solved knapsack feasibility problems *only one* component of y was a noninteger in $(0,1)$. However, for this noninteger y_ℓ the product $a_\ell y_\ell$ was an integer and furthermore $a^T y$ was exactly equal to c. This allowed us, as described in Algorithm 3, to either get an exact or approximate integer solution by setting $y_\ell = 0$. An exact solution was obtained when Algorithm 3 was able to find an a_i with $y_i = 0$ such that $a_i = a_\ell y_\ell$ and replacing $y_i = 0$ by $y_i = 1$. Otherwise, an approximate solution was obtained with a relative percentage error given by $100 * a_\ell y_\ell / c$.

6. The maximum average relative error percentage for the 10 approximate solutions, all with integer entries, was 0.03886%.

More details of the numerical results above are presented in Table 1. Test results for the second set of 1000 test problems are given in Table 2 for differently generated knapsack feasibility problems as follows. In particular, each component of the vector a of $a^T y = c$ is a positive integer picked from a uniform random distribution over the interval $[0,1000]$ instead of $[0,100]$. Furthermore a solution y to $a^T y = c$ was picked as a random n-dimensional binary integer vector from $\{0,1\}^n$ with approximately 25% of its components being zeros and 75% being ones instead of 50% each as is the case for problems in Table 1. The results are similar to those of Table 1 except that for $n \leq 5000$ the results of Table 1 are better.

5. Conclusion and Outlook

We have given an equivalent formulation to an integer feasibility problem as an absolute value equation in an n-dimensional real variable and presented a concave quadratic pro-

Table 1. Solution by Algorithm 3, utilizing the CPLEX linear programming code [3] within MATLAB [12], of 1000 consecutively generated random solvable knapsack feasibility problems $a^T y = c$, with each a_i randomly chosen from the set $\{1, 2, \ldots, 100\}$ and such that a solution y exists with approximately half of its components being zero. The times are for a 4 Gigabyte machine running Red Hat Enterprise Linux 5. Column 4 gives the number of instances where an exact solution was obtained at Step I of Algorithm 3, that is a global solution of the concave quadratic program (6) was achieved directly by the Successive Linearization Algorithm 1.

Problem Size n	100-Problem Avg Soln Time Sec	Total No. of Exact Solns Out of 100	No. of Exact Solns @ Step I Out of 100	Total No. of Approx Solns Out of 100	Avg Relative Error of Approx Solns
500	0.0389	97	15	3	0.03886%
1,000	0.0453	99	25	1	0.01982%
5,000	0.0779	98	41	2	0.001581%
10,000	0.1167	99	56	1	0.002775%
20,000	0.2166	98	52	2	0.0003928%
50,000	0.4343	99	66	1	0.00007920%
100,000	1.1609	100	65	0	—
200,000	2.0346	100	78	0	—
500,000	5.0359	100	82	0	—
1000,000	7.5911	100	95	0	—

gramming formulation that solves most of the randomly generated test problems. Interesting future work may involve the possibility of formulating *other* NP-hard integer feasibility or optimization problems as absolute value equations in real variables and coming up with linear or nonlinear programming formulations for their solution. Another interesting problem would be to find other efficient solution methods for the very simple absolute value equation (2).

Acknowledgments

Research described in this Data Mining Institute Report 08-03, September 2008, was supported by National Science Foundation Grant IIS-0511905.

Table 2. Solution by Algorithm 3, utilizing the CPLEX linear programming code [3] within MATLAB [12], of 1000 consecutively generated random solvable knapsack feasibility problems $a^T y = c$, with each a_i randomly chosen from the set $\{1,2,\ldots,1000\}$ and such that a solution y exists with approximately 25% of its components being zero and 75% are ones. The times are for a 4 Gigabyte machine running Red Hat Enterprise Linux 5. Column 4 gives the number of instances where an exact solution was obtained at Step I of Algorithm 3, that is a global solution of the concave quadratic program (6) was achieved directly by the Successive Linearization Algorithm 1.

Problem Size n	100-Problem Avg Soln Time Sec	Total No. of Exact Solns Out of 100	No. of Exact Solns @ Step I Out of 100	Total No. of Approx Solns Out of 100	Avg Relative Error of Approx Solns
500	0.1001	41	6	59	0.01002%
1,000	0.1188	59	6	41	0.003478%
5,000	0.2344	92	8	8	0.0001748%
10,000	0.3570	99	20	1	0.00008090%
20,000	0.6533	99	14	1	0.00003971%
50,000	1.5032	100	26	0	—
100,000	3.3187	100	35	0	—
200,000	7.0661	100	43	0	—
500,000	14.6279	100	60	0	—
1000,000	18.7115	100	81	0	—

References

1. S.-J. Chung. NP-completeness of the linear complementarity problem. *Journal of Optimization Theory and Applications*, 60:393–399, 1989.
2. R. Horst, P. Pardalos, and N. V. Thoai. *Introduction to Global Optimization*. Kluwer Academic Publishers, Dodrecht, Netherlands, 1995.
3. ILOG, Incline Village, Nevada. *ILOG CPLEX 9.0 User's Manual*, 2003. http://www.ilog.com/products/cplex/.
4. O. L. Mangasarian. *Nonlinear Programming*. McGraw–Hill, New York, 1969. Reprint: SIAM Classic in Applied Mathematics 10, 1994, Philadelphia.
5. O. L. Mangasarian. The linear complementarity problem as a separable bilinear program. *Journal of Global Optimization*, 6:153–161, 1995.
6. O. L. Mangasarian. Machine learning via polyhedral concave minimization. In H. Fischer, B. Riedmueller, and S. Schaeffler, editors, *Applied Mathematics and Parallel Computing - Festschrift for Klaus Ritter*, pages 175–188. Physica-Verlag A Springer-Verlag Company, Heidelberg, 1996. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/95-20.ps.
7. O. L. Mangasarian. Solution of general linear complementarity problems via nondifferentiable concave minimization. *Acta Mathematica Vietnamica*, 22(1):199–205, 1997. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/96-10.ps.
8. O. L. Mangasarian. Minimum-support solutions of polyhedral concave programs. *Optimization*, 45:149–162, 1999. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/97-05.ps.
9. O. L. Mangasarian. Absolute value equation solution via concave minimization. *Optimization Letters*, 1(1):3–8, 2007. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/06-02.pdf.

10. O. L. Mangasarian. A generlaized newton method for absolute value equations. Technical Report 08-01, Data Mining Institute, Computer Sciences Department, University of Wisconsin, May 2008. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/08-01.pdf. Optimization Letters, to appear.
11. O. L. Mangasarian and R. R. Meyer. Absolute value equations. *Linear Algebra and Its Applications*, 419:359–367, 2006. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/05-06.pdf.
12. MATLAB. *User's Guide*. The MathWorks, Inc., Natick, MA 01760, 1994-2006. http://www.mathworks.com.