Wu et al. Appl Inform (2017) 4:18
https://doi.org/10.1186/s40535-017-0047-0

Applied Informatics

RESEARCH Open Access

Solving the large-scale knapsack feasibility problem using a distributed computation approach to integer programming

Zhengtian Wu[1,2*], Fuyuan Hu[1] and Baochuan Fu[1]

*Correspondence: wzht8@mail.usts.edu.cn
[1] School of Electronic and Information Engineering, Suzhou University of Science and Technology, Kerui Road, Suzhou, China
Full list of author information is available at the end of the article

Abstract

The knapsack feasibility problems have been intensively studied both because of their immediate applications in industry and financial management, but more pronounced for theoretical reasons, as knapsack problems frequently occur by relaxation of various integer programming problems. In this work, the large-scale knapsack feasibility problem is divided into two subproblems. The first subproblem is transforming of the knapsack feasibility problem into a polytope judgement problem which is based on lattice basis reduction. In the next subproblem, a distributed implementation of Dang and Ye's fixed-point iterative algorithm is introduced to solve the polytope judgement problem generated in the former subproblem. Compared with the branch and bound method, numerical results show that this distributed fixed-point method is effective.

Keywords: The knapsack feasibility problems, Integer programming, Distributed implementation, Fixed-point iterative algorithm

Introduction

The knapsack problems, which have been intensively studied since the pioneering work of Dantzig (1957), have played an important role in industry, financial management, and so on. What is more, various integer programming problems can be relaxed to the knapsack problems. Therefore, the computing of the knapsack problems has been becoming the mark level of the computation in integer problems. To tackle this problem, many new contributions have been made in the following literature. A heuristic based upon genetic algorithms has been developed for multidimensional knapsack problem in paper (Chu and Beasley 1998). Based on the harmony search method, a new binary-coded version of harmony search (Kong et al. 2015) is presented to solve large-scale multidimensional knapsack problem. In this proposed algorithm, attention is paid to the probability distribution rather than the exact value of each decision variable, and the concept of mean harmony is developed in the memory consideration. Inspired by region partition of items, an effective hybrid algorithm based on greedy degree and expectation efficiency is constructed in the paper (Lv et al. 2016). Combining advanced features both from the path relinking method and the responsive threshold search algorithm, the first evolutionary path relinking approach is introduced in paper (Chen et al. 2016) for solving the quadratic multiple knapsack problem approximately.

In this paper, the distributed Dang and Ye's fixed-point iterative method (Dang and Ye 2015) is implemented to solve large-scale knapsack feasibility problem. This fixed-pointed algorithm has been extended to airline disruption problem and other problems. The idea of solving multiple fleet airline disruption problems using a distributed computation approach to integer programming has been developed in the previous work (Wu et al. 2017a). The paper (Wu et al. 2017b) uses the Dang and Ye's iterative fixed-point method for integer programming to generate feasible flight routes which are used to construct an aircraft reassignment in response to the grounding of one aircraft. Some computing of Nash equilibria methods are derived from this Dang and Ye's algorithm, such as the method from the papers (Wu et al. 2014, 2015). Some other engineering problems (Zhang et al. 2015, 2016) can also use this fixed -point method to solve. The Dang and Ye's fixed-pointed method can be explained as follows.

Let $\boldsymbol{P} = \{\boldsymbol{x} \in R^n | \boldsymbol{A}\boldsymbol{x} + \boldsymbol{G}\boldsymbol{w} \le \boldsymbol{b}$, for some $\boldsymbol{w} \in R^p\}$, where $\boldsymbol{A} \in R^{m \times n}$ is an $m \times n$ integer matrix with $n \ge 2$, $\boldsymbol{G} \in R^{m \times p}$ an $m \times p$ matrix, and $\boldsymbol{b}$ a vector of R^m.

Let $x^{\max} = (x_1^{\max}, x_2^{\max}, \ldots, x_n^{\max})^T$ with $x_j^{\max} = \max_{x \in P} x_j$, $j = 1, 2, \ldots, n$ and $x^{\min} = (x_1^{\min}, x_2^{\min}, \ldots, x_n^{\min})^T$ with $x_j^{\min} = \min_{x \in P} x_j$, $j = 1, 2, \ldots, n$. Let $D(P) = \left\{\boldsymbol{x} \in Z^n | \boldsymbol{x}^l \le \boldsymbol{x} \le \boldsymbol{x}^u\right\}$, where $\boldsymbol{x}^l = \lfloor \boldsymbol{x}^{\min} \rfloor$ and $\boldsymbol{x}^u = \lfloor \boldsymbol{x}^{\max} \rfloor$. For $\boldsymbol{z} \in R^n$ and $k \in N_0$, let $P(\boldsymbol{z}, k) = \{\boldsymbol{x} \in P | x_i = z_i,\ 1 \le i \le k,$ and $x_i \le z_i,\ k+1 \le i \le n\}$.

Given an integer point $y \in D(P)$ with $y_1 > x_i^l$, Dang and Ye (2015) developed a fixed-point iterative algorithm which is presented in Fig. 1.

In Dang and Ye's algorithm for integer problem, a definition of an increasing mapping, which is from a finite lattice into itself, is developed. Any integer point, which is outside the P, is mapped into the first point in P that is smaller than them in the lexicographical

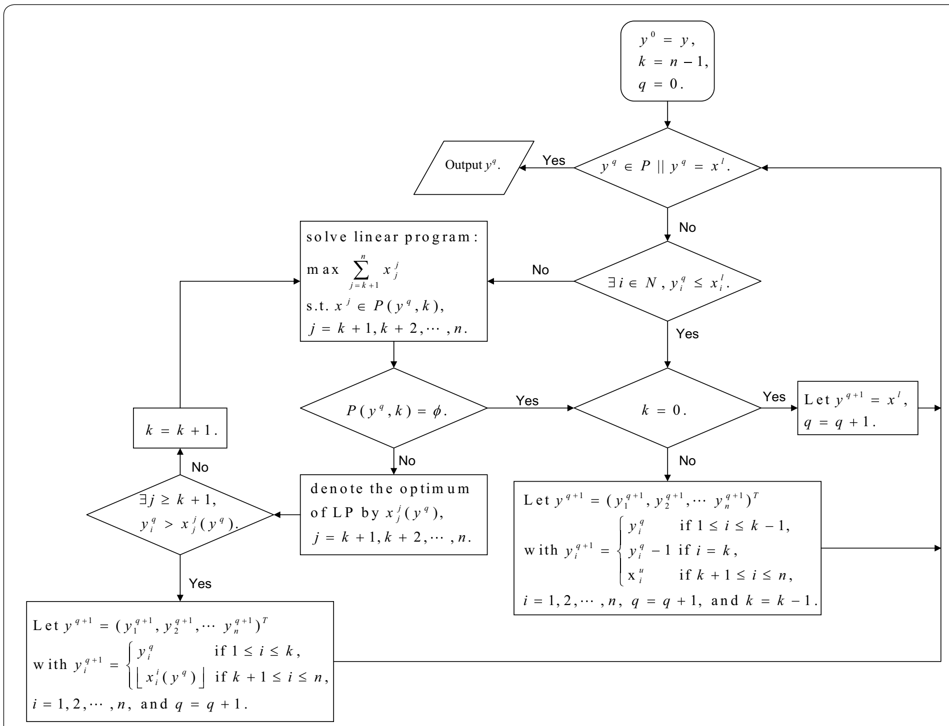

Fig. 1 Flow diagram of the iterative method

order of x^l. All the integer points, which are inside the polytope, are the fixed points through this increasing mapping. Given an initial integer point, the method either proves there is no such point by a limited number of iterations or generates an integer point in the polytope. For more proofs and details about this iterative algorithm, one can consult the paper (Dang and Ye 2015). An appeal feature of this fixed-point iterative method, which will be used in this paper, is that it can be easily implemented in a distributed way.

The rest of the paper is organized as follows. Some details of transformation of the knapsack feasibility problem into a polytope judgement problem, which is based on a LLL basis reduction, will be presented in "Transformation of the knapsack feasibility problem into apolytope judgement problem" section. The computation details and numerical results will be given in "Distributed computation and numerical results" section. Some conclusions and future work will be discussed in the last "Conclusions and future work" section.

Transformation of the knapsack feasibility problem into a polytope judgement problem

In order to analyze the knapsack feasibility problem easily, this problem is defined as follows (Dantzig 1957).

Definition 1 Find a 0–1 integer solution of $p^T x = d$, where $p = (p_1, p_2, ..., p_{n+1})^T > 0$ and $p_i \neq p_j$ for all $i \neq j$

After analysis, one can see this knapsack problem is one special example of the market split problem. To convert this problem into an equivalent problem of determining whether there is an integer point in a full-dimensional polytope given by $P = \{x \in R^n | Ax \leq b\}$, we can apply the LLL basis reduction algorithm (Khorramizadeh 2012) which has been described in paper (Wu et al. 2013).

Therefore, this knapsack feasibility problem can be formulated equivalently as follows: Does there exist a vector

$$\boldsymbol{\lambda} \in Z^{n-m} \text{ s.t.} - \boldsymbol{x}_d \leq \boldsymbol{X}_0 \boldsymbol{\lambda} \leq \boldsymbol{e}^{(n-m)\times 1} - \boldsymbol{x}_d. \quad (1)$$

The solving of the Definition 1 can be transformed to judge whether there exists an integer point in the polytope (1). The distributed iterative algorithm, developed by Dang and Ye (2015), has a good performance to judge whether the polytope (1) has an integer point or not.

Distributed computation and numerical results

One master computer combined with several slave computers is equipped to implement the distributed computation. The master computer takes charge of solving the solution space of the polytope, dividing the solution space to segments, sending the segments to the slave computers, receiving the computation result from the slave computers, and exporting the computation result. Each slave computer receives the segment, judges whether there exits an integer point in its segment using Dang and Ye's fixed-point iterative method, and sends its result to the master computer. The outline of the distributed computation process in this paper is illustrated in Fig. 2

In present study, the distributed computation system consists of three computers which are OptiPlex 330 with two processors. The programs are written in C++ language. Subsegments which are distributed by the master computer are mutually independent. Therefore, subsegments can be solved simultaneously. Luc Lapointe (1994) has been adopted to establish a communication network among the computers. The other settings and parameters are taken the same as the research (Wu et al. 2013). Message Passing Interface (MPI) is a tool to establish a stable communication network between the master computer and the slave computers in the distributed computation. The pseudocode of building the distributed network by MPI is described in Fig. 3.

In the following case study, two methods have been employed to solve the same example, respectively. One is distributed fixed-point method, while the other is the branch and cut method (Padberg and Rinaldi 1991) which is a classical algorithm for integer programming. The numerical results are shown in the following Table 1. For simplification, some symbols are defined as follows.

NumLPs: The number of iterations for a certain algorithm
BC: The branch and cut method

From Table 1, we can see that the distributed algorithm is superior to the branch and cut method regarding the number of iterations.

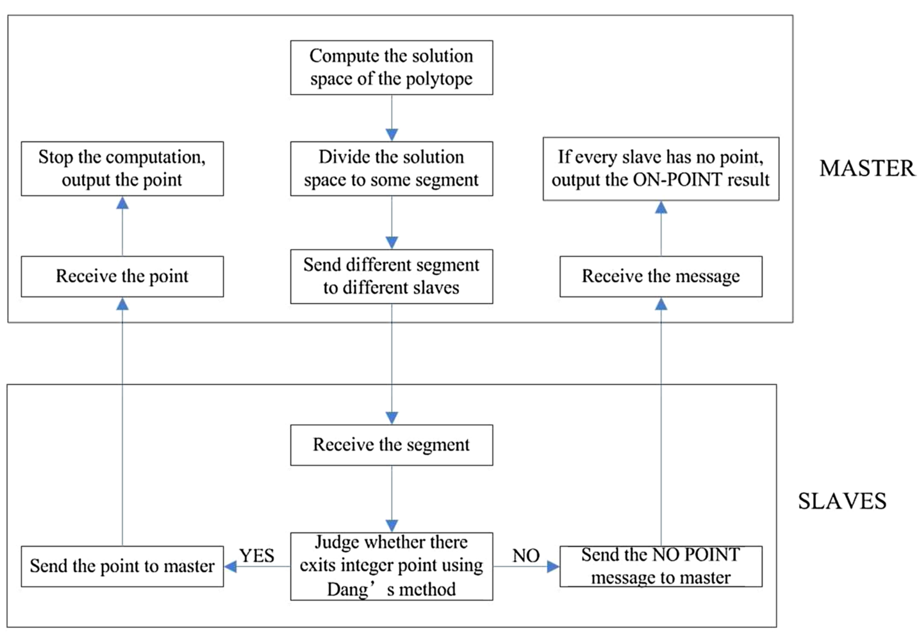

Fig. 2 Flow diagram of the distributed computation process

```
Distributed computation implementation

begin
MPI_Init(&argc,&argv);
MPI_Comm_rank(MPI_COMM,computerid)
GetSystemInfo(SystemInfo)
if(computerid = 0)
    begin
    receive SystemInfo.dwNumberOfProcessors from computerid ≠
0
    divide the solution space based on the total number of processors
and calculate x^{e_i} and x^{s_i} defining each segment
    send x^{e_i} and x^{s_i} to each computerid ≠ 0
    #pragma omp parallel
        for(i=0;i< SystemInfo.dwNumberOfProcessors;i++)
            begin
             compute feasible integer points in remaining segments by
CPLEX CP Optimizer or Dang and Ye's method
            end
    receive feasible integer points from each computerid ≠ 0
    end
else
    begin
    send SystemInfo.dwNumberOfProcessors to computerid = 0
    receive x^{e_i} and x^{s_i} from computerid = 0
    #pragma omp parallel
        for(i=0;i¡SystemInfo.dwNumberOfProcessors;i++)
            begin
             compute feasible integer points in received segments by
CPLEX CP Optimizer or Dang and Ye's method
            end
    send feasible integer points to computerid = 0
    end
end
```

Fig. 3 Distributed computation pseudocode

Table 1 Two method to solve the knapsack feasibility problems

Prob.	Dimension n	The method		BC	
		NumLPs	F	NumLPs	F
1	1000	1011	Feasible	1428	Feasible
2	1000	1035	Feasible	2023	Feasible
3	1000	1002	Feasible	1360	Feasible
4	1000	1003	Feasible	1117	Feasible
5	1000	1005	Feasible	1122	Feasible
6	1000	999	Infeasible	1315	Infeasible
7	1000	1002	Feasible	978	Feasible
8	1000	1007	Feasible	1486	Feasible
9	1000	1065	Feasible	2017	Feasible
10	1000	1016	Feasible	2587	Feasible
11	1000	1014	Feasible	879	Feasible
12	1000	1012	Feasible	724	Feasible

Conclusions and future work

In this paper, a new method has been used to solve the knapsack feasibility problem. This method is divided into two steps. In the first step, the knapsack feasibility problem is transformed into a polytope judgement problem based on a LLL basis reduction. In the other step, a distributed fixed-point method for integer programming is implemented to solve the polytope judgement problem. Compared with the branch and cut method which is considered to be the best algorithm for the problem of this kind, numerical results show that this distributed fixed-point method is promising.

However, the dimension of instances is low and the number of slave computers in the numerical experiment is only three. The potential ability of this method has not been fully expressed. In the next step, these two shortcomings will be settled. With large number of slave computers, one can be confident to believe that the numerical results will be more better. Additionally, the distributed method is easy to be extended to solving other problems.

Authors' contributions

The authors discussed the problem and the solutions were proposed all together. All authors participated in drafting and revising the final manuscript. All authors read and approved the final manuscript.

Author details

[1] School of Electronic and Information Engineering, Suzhou University of Science and Technology, Kerui Road, Suzhou, China. [2] Department of Systems Engineering and Engineering Management, City University of Hong Kong, Tat Chee Avenue, Hong Kong, China.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Funding

This work was partly supported by National Nature Science Foundation of China under Grants 71471091, 71271119, and 61672371; Research Foundation of USTS under Grants No. XKQ201517; Jiangsu Provincial Department of Housing and Urban-Rural Development under Grants No. 2017ZD253; Natural science fund for colleges and universities in Jiangsu Province under Grants No.17KJD110008; Fundamental Research Funds for the Central Universities under Grant No.30917011339; and Natural Science Foundation of Jiangsu Province under Grant No. BK20170820.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 28 August 2017 Accepted: 14 December 2017
Published online: 20 December 2017

References

Chen Y, Hao JK, Glover F (2016) An evolutionary path relinking approach for the quadratic multiple knapsack problem. Knowl Based Syst 92:23–34

Chu PC, Beasley JE (1998) A genetic algorithm for the multidimensional knapsack problem. J Heuristics 4(1):63–86

Dang C, Ye Y (2015) A fixed point iterative approach to integer programming and its distributed computation. Fixed Point Theory Appl 2015(1):1–15

Dantzig GB (1957) Discrete-variable extremum problems. Oper Res 5(2):266–288

Khorramizadeh M (2012) Numerical experiments with the lll-based hermite normal form algorithm for solving linear diophantine systems. Int J Contemp Math Sci 7(13):599–613

Kong X, Gao L, Ouyang H, Li S (2015) Solving large-scale multidimensional knapsack problems with a new binary harmony search algorithm. Comput Oper Res 63:7–22

Lv J, Wang X, Huang M, Cheng H, Li F (2016) Solving 0–1 knapsack problem by greedy degree and expectation efficiency. Appl Soft Comput 41:94–103

Luc Lapointe JM (1994) Message Passing Interface Forum. MPI: A Message Passing Interface Standard. J combin theory Ser A 112:44–81

Padberg M, Rinaldi G (1991) A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. SIAM Rev 33(1):60–100

Wu Z, Dang C, Zhu C (2013) Solving the market split problem using a distributed computation approach. In: 2013 IEEE international conference on information and automation (ICIA), IEEE, pp 1252– 1257

Wu Z, Dang C, Karimi HR, Zhu C, Gao Q (2014) A mixed 0-1 linear programming approach to the computation of all pure-strategy nash equilibria of a finite n-person game in normal form. Math Probl Eng 2014:8. https://doi.org/10.1155/2014/640960

Wu Z, Dang C, Hu F, Fu B (2015) A new method to finding all Nash equilibria. In: International conference on intelligent science and big data engineering, pp 499–507

Wu Z, Li B, Dang C (2017a) Solving multiple fleet airline disruption problems using a distributed-computation approach to integer programming. IEEE Access 5:19116–19131

Wu Z, Li B, Dang C, Hu F, Zhu Q, Fu B (2017b) Solving long haul airline disruption problem caused by groundings using a distributed fixed-point computational approach to integer programming. Neurocomputing 269:232–255. https://doi.org/10.1016/j.neucom.2017.02.091
Zhang Y, Lei ZX, Zhang LW, Liew KM, Yu JL (2015) Nonlocal continuum model for vibration of single-layered graphene sheets based on the element-free kp-Ritz method. Eng Anal Bound Elem 56:90–97
Zhang Y, Zhang LW, Liew KM, Yu JL (2016) Free vibration analysis of bilayer graphene sheets subjected to in-plane magnetic fields. Compos Struct 144:86–95