



ELSEVIER

European Journal of Operational Research 140 (2002) 584–589

EUROPEAN
JOURNAL
OF OPERATIONAL
RESEARCH

www.elsevier.com/locate/dsw

Discrete Optimization

Approximately global optimization for assortment problems using piecewise linearization techniques

Han-Lin Li ^{a,*}, Ching-Ter Chang ^b, Jung-Fa Tsai ^a

^a Institute of Information Management, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu 30050, Taiwan, ROC

^b Business Department of Education, National Changhua University of Education, Hsinchu 30050, Taiwan, ROC

Received 25 January 2000; accepted 23 May 2001

Abstract

Recently, Li and Chang proposed an approximate model for assortment problems. Although their model is quite promising to find approximately global solution, too many 0–1 variables are required in their solution process. This paper proposes another way for solving the same problem. The proposed method uses iteratively a technique of piecewise linearization of the quadratic objective function. Numerical examples demonstrate that the proposed method is computationally more efficient than the Li and Chang method. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Cutting; Assortment; Global optimization

1. Introduction

Assortment problems occur when a number of small rectangular pieces need to be cut from a large rectangle to get minimum area. Recently, Li and Chang [1] developed a method for finding the global optimal solution of the assortment problem. Li and Chang's method, however, requires to use numerous 0–1 variables to linearize the polynomial objective function in their model, which would cause heavy computational burden. This paper proposes instead a piecewise linearization method. The major advantage of this method is that it uses

much less number of 0–1 variables to linearize the quadratic objective function than used in Li and Chang's model. The computational efficiency can therefore be improved significantly. The numerical examples demonstrate that the computation time of the proposed method is much less than that in Li and Chang's model.

2. Problem formulation

Given n rectangles with fixed lengths and widths. An assortment optimization problem is to allocate all of these rectangles within an enveloping rectangle, which has minimum area. Denote x and y as the width and the length of the enveloping rectangle ($x > 0$, $y > 0$), the assortment optimization problem is stated briefly as follows:

* Corresponding author. Tel.: +886-3-5728709; fax: +886-3-5723792.

E-mail address: hlli@cc.nctu.edu.tw (H.-L. Li).

Minimize xy

subject to

1. All of n rectangles are non-overlapping.
2. All of n rectangles are within the range of x and y .
3. $0 < x \leq \bar{x}$ and $0 < y \leq \bar{y}$ (\bar{x} and \bar{y} are constants).

An assortment optimization problem can be formulated below. The related notations are those of Li and Chang [1].

2.1. Assortment problem

Minimize xy

subject to

$$(x'_i - x'_k) + u_{ik}\bar{x} + v_{ik}\bar{x} \geq \frac{1}{2}[p_i s_i + q_i(1 - s_i) + p_k s_k + q_k(1 - s_k)] \quad \forall i, k \in J, \tag{2}$$

$$(x'_k - x'_i) + (1 - u_{ik})\bar{x} + v_{ik}\bar{x} \geq \frac{1}{2}[p_i s_i + q_i(1 - s_i) + p_k s_k + q_k(1 - s_k)] \quad \forall i, k \in J, \tag{3}$$

$$(y'_i - y'_k) + u_{ik}\bar{y} + (1 - v_{ik})\bar{y} \geq \frac{1}{2}[p_i(1 - s_i) + q_i s_i + p_k(1 - s_k) + q_k s_k] \quad \forall i, k \in J, \tag{4}$$

$$(y'_k - y'_i) + (1 - u_{ik})\bar{y} + (1 - v_{ik})\bar{y} \geq \frac{1}{2}[p_i(1 - s_i) + q_i s_i + p_k(1 - s_k) + q_k s_k] \quad \forall i, k \in J, \tag{5}$$

$$\bar{x} \geq x \geq x'_i + \frac{1}{2}[p_i s_i + q_i(1 - s_i)] \quad \forall i \in J, \tag{6}$$

$$\bar{y} \geq y \geq y'_i + \frac{1}{2}[p_i(1 - s_i) + q_i s_i] \quad \forall i \in J, \tag{7}$$

$$x'_i - \frac{1}{2}[p_i s_i + q_i(1 - s_i)] \geq 0 \quad \forall i \in J, \tag{8}$$

$$y'_i - \frac{1}{2}[p_i(1 - s_i) + q_i s_i] \geq 0 \quad \forall i \in J, \tag{9}$$

where u_{ik}, v_{ik}, s_i, s_k are 0–1 variables, and x, y, x'_i, x'_k, y'_i and y'_k are bounded continuous variables, p_i and q_i are length and width of i th object.

Constraints (2)–(5) are the non-overlapping conditions and constraints (6)–(9) ensure that all rectangles are within the enveloping rectangle.

Li and Chang [1] proposed an approach for solving this problem to obtain a global optimum. The basic idea of their method is to approximately

substitute x and y continuous variables in (1) by a set of 0–1 variables thus to linearize the product term xy . Problem (1)–(9) can then be reformulated as a linear mixed 0–1 problem which can be solved to reach a global optimum within a tolerable error.

3. Li and Chang approach

Li and Chang [1] substitute x and y by the following 0–1 representation:

$$x = \bar{\epsilon}_x \sum_{g=1}^G 2^{g-1} \theta_g + \epsilon_x, \tag{10}$$

$$y = \bar{\epsilon}_y \sum_{h=1}^H 2^{h-1} \delta_h + \epsilon_y, \tag{11}$$

where ϵ_x and ϵ_y are small positive variables. $\bar{\epsilon}_x$ and $\bar{\epsilon}_y$ are the pre-specified constants which are the upper bounds of ϵ_x and ϵ_y , respectively. θ_g and δ_h are 0–1 variables, and G and H are integers which denote the number of required 0–1 variables for representing x and y .

Then the Li and Chang model is reformulated as a linear mixed 0–1 program below.

3.1. Model 1 [1]

$$\text{Minimize } \bar{\epsilon}_x \sum_{g=1}^G 2^{g-1} z_g + \bar{\epsilon}_y \sum_{h=1}^H 2^{h-1} u_h \tag{12}$$

subject to

$$z_g \geq y + \bar{y}(\theta_g - 1), \quad g = 1, 2, \dots, G, \tag{13}$$

$$u_h \geq \epsilon_x + \bar{\epsilon}_x(\delta_h - 1), \quad h = 1, 2, \dots, H, \tag{14}$$

constraints in (2)–(9)

$$z_g \geq 0, \quad u_h \geq 0, \quad \theta_g, \delta_h \in \{0, 1\}. \tag{15}$$

The major difficulty of Model 1 is that it involves $G + H$ additional 0–1 variables. The smaller the tolerable errors (i.e., ϵ_x and ϵ_y), the larger the size of G and H and the longer the CPU time for solving the problem.

4. Proposed new linear strategy

This paper proposes another strategy for linearizing the quadratic objective function xy in (1).

Define K be a set expressed as

$$K = \{0 < \underline{x} \leq x \leq \bar{x}, 0 < \underline{y} \leq y \leq \bar{y}, x, y \in F, \\ F \text{ is a feasible set}\}.$$

It is clear that an optimization program P1: $\{\text{Minimize Obj1} = xy | x, y \in K\}$ is equivalent to the program below.

$$P2 : \{\text{Minimize Obj2} = \ln x + \ln y | x, y \in K\}.$$

Following propositions discuss the proposed approach of linearizing the logarithmic terms $\ln x$ and $\ln y$.

Proposition 1 [3]. *A logarithm function $\ln x$, $0 < a_1 \leq x \leq a_m$, as shown in Fig. 1, can piecewise linearly be approximated*

$$\ln x \doteq \ln \hat{x} = \ln a_1 + s_1(x - a_1) \\ + \sum_{j=2}^{m-1} \frac{s_j - s_{j-1}}{2} (|x - a_j| + x - a_j), \tag{16}$$

where $a_j, j = 1, 2, \dots, m$, are the break points of $\ln x$, $a_j < a_{j+1}$ and s_j are the slopes of line segments between a_j and a_{j+1} ,

$$s_j = \frac{\ln a_{j+1} - \ln a_j}{a_{j+1} - a_j} \quad \text{for } j = 1, 2, \dots, m - 1.$$

Proposition 2. *Since $\ln x$ is concave function, it is clear that the approximation bounds $\ln x$ from below. That means $\ln x \geq \ln \hat{x}$.*

Proposition 3 (Lower bound). *Consider the following two programs:*

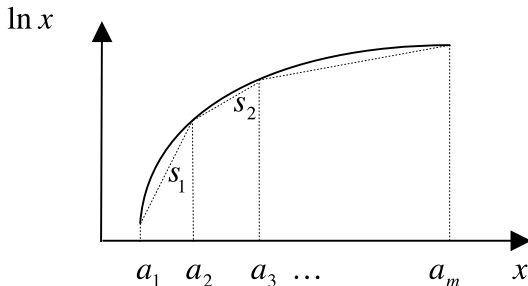


Fig. 1. Piecewise linearization of $\ln x$.

$$P2 : \{\text{Minimize Obj2} = \ln x + \ln y | x, y \in K\}.$$

$$P3 : \{\text{Minimize Obj3} = \ln \hat{x} + \ln \hat{y} | x, y \in K\}.$$

Program P3 provides a lower bound on Program P2 due to Proposition 2.

Now we discuss the way to linearize $\ln \hat{x}$. Consider the following proposition.

Proposition 4 (Linearization). *$\ln \hat{x}$ in (16) can be linearized as follows.*

$$\ln \hat{x} = \ln a_1 + s_1(x - a_1) \\ + \sum_{j=2}^{m-1} (s_j - s_{j-1})(a_j u_j + x - a_j - w_j) \tag{17}$$

where

- (i) $-a_m u_j \leq x - a_j \leq a_m(1 - u_j)$
for $j = 2, 3, \dots, m$,
- (ii) $-a_m u_j \leq w_j \leq a_m u_j$
for $j = 2, 3, \dots, m$,
- (iii) $a_m(u_j - 1) + x \leq w_j \leq a_m(1 - u_j) + x$
for $j = 2, 3, \dots, m$,
- (iv) $u_j \geq u_{j-1}$ for $j = 2, 3, \dots, m$,
- (v) u_j are 0–1 variables and $w_j \geq 0$
for $j = 2, 3, \dots, m$.

Proof. If $x - a_j \geq 0$ then $u_j = 0$ and $w_j = 0$ based on (i) and (ii); which results in

$$a_j u_j + x - a_j - w_j = (|x - a_j| + x - a_j) / 2 \\ = x - a_j.$$

If $x - a_j < 0$ then $u_j = 1$ and $w_j = x$ based on (i) and (iii); which results in

$$a_j u_j + x - a_j - w_j = (|x - a_j| + x - a_j) / 2 = 0.$$

Therefore, $\ln \hat{x}$ in (16) is equivalent to (17). Now we consider condition (iv).

Since $a_{j-1} < a_j$, if $x < a_j$ (i.e. $u_j = 1$) then $x < a_{j+1}$ and we have $u_{j+1} = 1$.

If $x > a_{j+1}$ (i.e. $u_{j+1} = 0$) then $x > a_j$, and we have $u_j = 0$.

Therefore, it is true that $u_j \geq u_{j-1}$, for $j = 2, 3, \dots, m$.

Condition (iv) is used to accelerate the computational speed of solving the problem. \square

5. Solution algorithm

From the above discussion, the proposed algorithm is as follows: Let S_r and T_r be respectively a set of break points of $\ln x$ and $\ln y$ at the r th iteration. Denote ε as a tolerable error. Then the linearization (17) is built up iteratively as follows.

Let $S_r = S_{r-1} \cup \{x(r)\}$, $T_r = T_{r-1} \cup \{y(r)\}$, where ‘ \cup ’ means union. Denote the number of elements in S_r as m_r , and the number of elements in T_r as n_r . Solving the following linear mixed 0–1 program:

$$\begin{aligned} & \text{Minimize } \text{Obj}(x(r+1)) + \text{Obj}(y(r+1)) \\ & \quad \text{(x,y)} \\ & = \ln a_1 + s_1(x - a_1) \\ & \quad + \sum_{j=2}^{m_r-1} (s_j - s_{j-1})(a_j u_j + x - a_j - w_j) \\ & \quad + \ln b_1 + t_1(y - b_1) \\ & \quad + \sum_{j=2}^{n_r-1} (t_j - t_{j-1})(b_j v_j + y - b_j - q_j) \end{aligned}$$

subject to

Restrictions part 1:

$$\begin{aligned} & -\bar{x}u_j \leq x - a_j \leq \bar{x}(1 - u_j), \\ & -\bar{x}u_j \leq w_j \leq \bar{x}u_j, \\ & \bar{x}(u_j - 1) + x \leq w_j \leq \bar{x}(1 - u_j) + x, \\ & u_j \geq u_{j-1}, \\ & \text{for } j = 2, 3, \dots, m_r \text{ (variable } x), \\ & a_1, a_2, \dots, a_{m_r} \in S_r, \\ & a_1 = \underline{x} < a_2 < \dots < a_{m_r} = \bar{x}; \end{aligned}$$

Restrictions part 2:

$$\begin{aligned} & -\bar{y}v_j \leq y - b_j \leq \bar{y}(1 - v_j), \\ & -\bar{y}v_j \leq q_j \leq \bar{y}v_j, \\ & \bar{y}(v_j - 1) + y \leq q_j \leq \bar{y}(1 - v_j) + y, \\ & v_j \geq v_{j-1}, \\ & \text{for } j = 2, 3, \dots, n_r \text{ (variable } y), \\ & b_1, b_2, \dots, b_{n_r} \in T_r, \\ & b_1 = \underline{y} < b_2 < \dots < b_{n_r} = \bar{y}; \end{aligned}$$

u_j, v_j are 0–1 variables, $w_j, q_j \geq 0$.

Restrictions part 3:

constraints in (2)–(9).

Let the solution be $(x(r+1), y(r+1))$.

If $|\text{Obj}(x(r+1)) - \ln x(r+1)| < \varepsilon$ and $|\text{Obj}(y(r+1)) - \ln y(r+1)| < \varepsilon$ then terminate the process, and $(x(r+1), y(r+1))$ is the optimal solution.

Otherwise, let $r = r + 1$ and resolve the above program. (A precise procedure is available by request, etc. from the authors.)

Proposition 5 (Convergence). *The above algorithm (run with $\varepsilon=0$) terminates with the incumbent solution (\hat{x}^*, \hat{y}^*) being optimum to the assortment problem (1)–(9) when $r \rightarrow \infty$.*

Proof. By the concavity of $\ln x$ (and $\ln y$) in (16) and the mean value theorem, $\ln \hat{x}^*$ (and $\ln \hat{y}^*$) are the lower bounds of $\ln x$ (and $\ln y$); (\hat{x}^*, \hat{y}^*) therefore is the optimal solution. \square

6. Numerical examples

Consider the following assortment optimization problems adopted from Li and Chang [1]: Some given rectangles are required to be placed within a rectangle which has minimum area. The sizes of pieces of rectangles are given in Table 1. Here we solve the same problem using Model 1 [1] and proposed model by LINGO [2], a common-used optimization package, running in a Pentium III 1000 personal computer.

Model 1 solves problem 1 by specifying $\bar{e}_x = \bar{e}_y = 0.1$, and obtains the global optimal solution $(x, y) = (31, 38)$ with the objective value 1178. Proposed model solves Problem 1 by specifying $\bar{e}_x = \bar{e}_y = 0.1$ and obtains the same solution as found by Model 1. Table 1 shows that for Problem 1 containing four rectangles, the proposed model only spends 1/7 of CPU time as spent in Model 1. For Problem 2 containing five rectangles, the proposed model uses much less CPU time than Model 1. The associated graphs are presented in Figs. 2 and 3.

To compare the capability of the two models in treating larger sizes of assortment problems, two

Table 1
Computational comparison of models with rectangles

Problem	Number of rectangles	p_i	q_i	CPU time (hh:mm:ss)		Objective value	
				Model 1	Proposed model	Model 1	Proposed model
Problem 1	4	24	20	00:05:12	00:00:44	1178 Global optimum	1178 Global optimum
		18	16				
		16	14				
		21	7				
Problem 2	5	33	10	>10:00:00	00:16:16	NA	1518 Global optimum
		30	11				
		25	15				
		18	14				
		18	10				

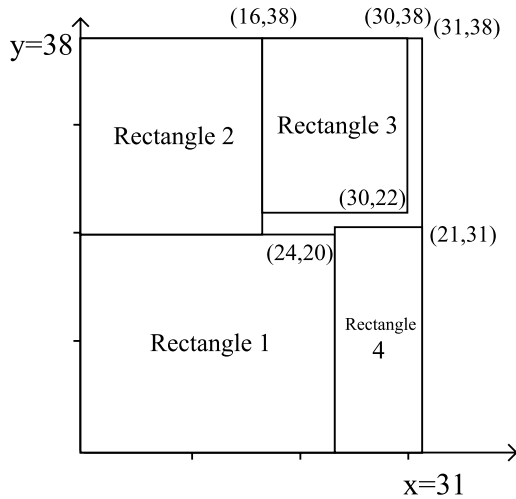


Fig. 2. Result for Problem 1 (four rectangles).

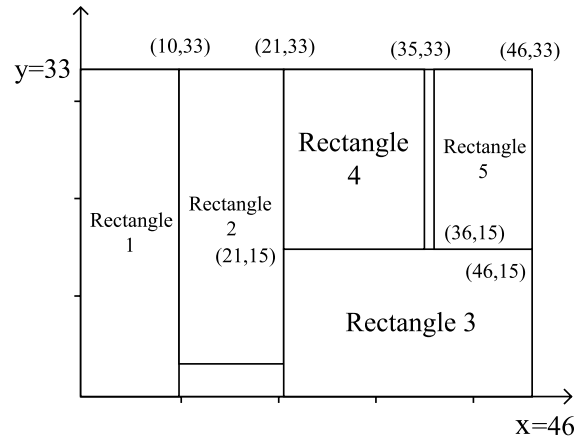


Fig. 3. Result for Problem 2 (five rectangles).

Table 2
Computational comparison of models with squares

Problem	Number of squares	Side p_i	CPU time (hh:mm:ss)		Objective value	
			Model 1	Proposed model	Model 1	Proposed model
Problem 3	4	1×1	00:04:48	00:00:37	25 Global optimum	25 Global optimum
	3	2×2				
	1	3×3				
Problem 4	4	1×1	>10:00:00	00:08:26	NA	30 Global optimum
	4	2×2				
	1	3×3				

other problems are examined as shown in Table 2. All rectangles in these two problems are squares. For Problem 3 where the number of squares is 8,

the proposed model spends less than 1/7 CPU time as spent in Model 1 to obtain a global optimum. The result is depicted in Fig. 4. For Problem 4 with

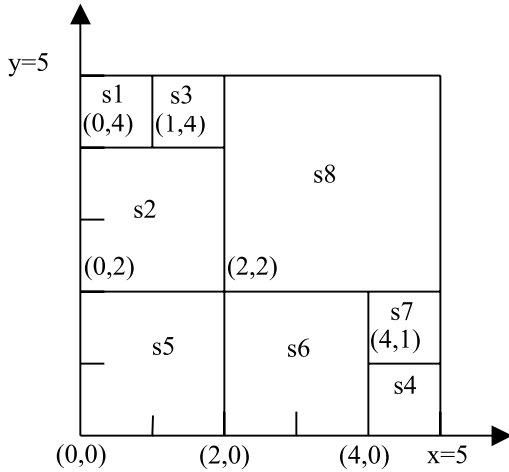


Fig. 4. Result for Problem 3 (eight squares).

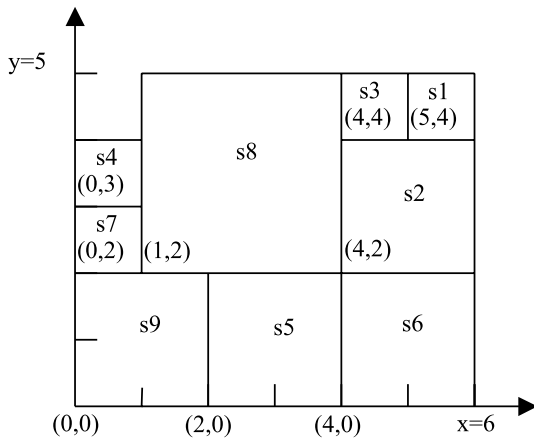


Fig. 5. Result for Problem 4 (nine squares).

nine squares, Model 1 cannot find the solution within 10 hours while the proposed model takes eight and half minutes to find the global solution. The solution is depicted in Fig. 5. The reported CPU times can be improved if these problems are run in a workstation computer instead of running in a personal computer.

7. Conclusions

This paper proposes a piecewise linearization method to solve the assortment problem. By piecewisely linearizing the quadratic objective function in the assortment problem, the proposed method reformulates the original problem as a linear mixed 0–1 program. Solving the linear mixed 0–1 problem iteratively, the proposed method can finally find a global optimum. Numerical examples demonstrate that the proposed method uses much less CPU time than that in [1] for reaching the global optimum.

References

- [1] H.L. Li, C.T. Chang, An approximately global optimization method for assortment problems, *European Journal of Operational Research* 105 (1998) 604–612.
- [2] LINGO Hyper 6.0: Lindo Systems Inc., 1999.
- [3] H.L. Li, C.S. Yu, A fuzzy multiobjective program with quasiconcave membership functions and fuzzy coefficients, *Fuzzy Sets and Systems* 109 (2000) 59–81.