Tạp chí Tin học và Điều khiển học, T.25, S.3 (2009), 214–230

MỘT THUẬT TOÁN LAI GHÉP GIẢI BÀI TOÁN CẮT VẬT TƯ MỘT CHIỀU VỚI NHIỀU KÍCH THƯỚC VẬT LIỆU THÔ

PHAN THỊ HOÀI PHƯƠNG[1], LƯƠNG CHI MAI[2], NGUYỄN VĂN HÙNG[2]

[1] *Học viện Công nghệ Bưu chính Viễn thông*

[2] *Viện Công nghệ thông tin, Viện Khoa học và Công nghệ Việt Nam*

Abstract. In this paper we study the method for solving One-Dimensional Cutting Stock Problem with Multiple Stock Sizes. Based on the decomposition of initial problem into the set of subproblems, our algorithm is built by hybridizing Genetic Algorithm and Column Generation method, in which the global optimal solution to the initial problem is searched by Genetic Algorithm from the set of local optimal solutions to the subproblems provided by Column Generation Method.

Tóm tắt. Trong bài báo này chúng ta sẽ nghiên cứu phương pháp giải bài toán cắt vật tư một chiều với nhiều kích thước vật liệu thô. Trên cơ sở phân rã bài toán ban đầu thành tập các bài toán nhỏ, thuật toán mới được xây dựng dựa trên sự lai ghép giữa thuật toán gen và thuật toán tạo sinh cột trong đó nghiệm tối ưu toàn cục sẽ được tìm kiếm bởi thuật toán gen trên cơ sở các tối ưu cục bộ do thuật toán tạo sinh cột mang lại khi giải các bài toán nhỏ.

1. GIỚI THIỆU

Bài toán cắt vật tư và xếp hộp (Cutting & Packing Problems) lần đầu tiên được Kantorovich hình thức hóa dưới dạng bài toán quy hoạch nguyên vào năm 1939 [22]. Trải qua 70 năm phát triển, mô hình đã thu hút mối quan tâm của đông đảo các nhà nghiên cứu cả về khía cạnh lý thuyết cũng như ứng dụng.

Về mặt lý thuyết, có hai đóng góp quan trọng làm cơ sở cho sự phát triển các phương pháp giải bài toán. Dantzig và Wolfe [28] đề xuất mô hình mới với tên gọi Dantzig-Wolfe Decomposition để khắc phục nhược điểm về suy biến liên tục yếu và tính đối xứng trong mô hình của Kantorovich. Độc lập với Dantzig và Wolfe, Gilmore và Gomory lần đầu tiên đưa kỹ thuật tạo sinh cột (Column Generation) do Ford và Fulkerson đề xuất (và sau này Dantzig và Wolfe hoàn thiện) - phương pháp được áp dụng rộng rãi và hiệu quả nhất khi giải các bài toán quy hoạch tuyến tính cỡ lớn - vào giải bài toán cắt vật tư [20,21]. Việc tăng tốc độ hội tụ của kỹ thuật tạo sinh cột trong việc giải bài toán cắt vật tư cũng được nhiều tác giả quan tâm [19].

Về khía cạnh ứng dụng, có rất nhiều biến thể khác nhau của bài toán được nghiên cứu trong thực tế $[1, 6-13, 15-26]$.

Bỏ qua những đặc điểm riêng của từng biến thể , nhiều tác giả đã hệ thống hóa chúng thành những lớp bài toán theo đặc trưng của cấu trúc logic làm cơ sở định hướng cho các nghiên cứu lý thuyết cũng như ứng dụng sau này $[2-5]$.

Trong các lớp bài toán cắt vật tư và xếp hộp, bài toán cắt vật tư một chiều (One-Dimensional Cutting Stock Problem - 1DCSP) có một vị trí quan trọng với số lượng công trình liên quan được công bố chiếm hơn một phần ba tổng số các công trình thuộc về chủ đề này [4]. Về mặt lý thuyết, bài toán được chỉ ra thuộc lớp NP-hard nên không tồn tại một thuật toán giải phổ quát với thời gian đa thức [27]. Chính vì vậy các phương pháp heuristics trở thành công cụ hữu hiệu khi nghiên cứu xây dựng giải pháp cho các bài toán thực tế. Trong [6], các tác giả đề xuất áp dụng thuật toán gen để giải bài toán cắt vật tư một chiều và chỉ ra tiềm năng của thuật toán trong việc tiết kiệm vật tư. Bài toán cắt vật tư cũng có thể được xét trong mối liên quan với các bài toán khác nhằm giảm chi phí tổng thể như kết hợp với bài toán vận tải [12], bài toán giá lắp đặt thiết bị cắt [10, 24, 26].

Bài toán cắt vật tư một chiều với nhiều kích thước vật liệu thô (One-Dimensional Cutting Stock Problem with Multiple Stock Sizes 1DMCSP) cũng được nhiều tác giả đề cập tới trong thời gian gần đây [8,9,10,16,30]. Bài toán 1DMCSP cũng trở nên công cụ hữu hiệu để giải quyết những thực tế nảy sinh trong sản xuất khi phế thải của bài toán cắt vật tư đối với một đơn hàng lại có thể sử dụng làm vật tư đầu vào cho các đơn hàng khác với một số điều kiện nhất định. Trong ngành xây dựng, bài toán có tính đến việc sử dụng phế liệu thép có đường kính lớn thay thế cho yêu cầu của các loại thép có đường kính nhỏ hơn nhằm giảm chi phí được đề cập tới trong [7]. Trong [11], các tác giả đề xuất kết hợp các phương pháp Modified First Fit Decreasing, Integer Programming và Data Mining để giải bài toán với yêu cầu bề rộng phế thải phải đủ lớn để có thể sử dụng trong tương lai.

Bài báo này dành cho việc xây dựng thuật toán lai ghép giữa thuật toán gen và thuật toán tạo sinh cột để giải bài toán 1DMCSP. Phần còn lại được được cấu trúc như sau: Mục 2 trình bày tóm tắt ý tưởng chính của kỹ thuật tạo sinh cột trong việc giải bài toán quy hoạch tuyến tính cỡ lớn và áp dụng để giải bài toán 1DCSP; Phương pháp phát biểu mới của bài toán 1DMCSP, một số kết quả lý thuyết làm cơ sở cho việc xây dựng thuật toán lai ghép thuật toán gen và kỹ thuật tạo sinh cột để giải bài toán được trình bày trong mục 3; Các kết quả thực tế được mô tả trong mục 4; Mục 5 là một số kết luận.

2. KỸ THUẬT TẠO SINH CỘT VÀ BÀI TOÁN CẮT VẬT TƯ MỘT CHIỀU

2.1. Kỹ thuật tạo sinh cột giải bài toán quy hoạch tuyến tính cỡ lớn

Phần này sẽ trình bày một số kết quả kinh điển của quy hoạch tuyến tính làm cơ sở cho việc phát triển nội dung của các phần tiếp theo.

Một bài toán quy hoạch tuyến tính có thể phát biểu như sau:

$$z_{LP} = \min\{cx | Ax \geq b,\ x \in R^n_+\}, \tag{1}$$

ở đây c là vector hàng n chiều giá trị thực được gọi là vector giá, A là ma trận thực kích thước $m \times n$ và b là vector cột m chiều giá trị thực được gọi là vector ràng buộc.

Bài toán quy hoạch tuyến tính có dạng:

$$z_{LP} = \min\{cx|Ax = b,\ x \in R^n_+\}, \tag{2}$$

được gọi là bài toán có dạng chuẩn.

Một trong các khái niệm cơ bản của quy hoạch tuyến tính là khái niệm đối ngẫu. Khi gọi bài toán quy hoạch tuyến tính cho trong (1) là bài toán nguyên thủy thì bài toán đối ngẫu với nó là bài toán quy hoạch tuyến tính được xác định như sau:

$$w_{LP} = \max\{ub|uA \leq c,\ u \in R^n_+\}. \tag{3}$$

Để giải bài toán quy hoạch tuyến tính (2), người ta sử dụng phương pháp đơn hình với mô tả tóm tắt như sau.

Cho trước bài toán quy hoạch tuyến tính dưới dạng chuẩn (2) với cơ sở là tập m vector cột độc lập tuyến tính bất kỳ của ma trận ràng buộc A. Không mất tính tổng quát ta có thể giả thiết đó là m vector cột đầu tiên của A và ta có thể biểu diễn A dưới dạng $A = BA'$. Sử dụng phương pháp Gauss để giải hệ phương trình:

$$BA'x = b.$$

Ta sẽ nhận được một nghiệm cơ sở của (2). Ma trận B khi đó được gọi là ma trận cơ sở và các biến tương ứng với ma trận B được gọi là các biến cơ sở. Giả sử c_B là vector giá của các biến cơ sở. Khi đó đối với mỗi j ta có thể xác định giá của biến x_j như sau:

$$\overline{c}_j = c_j - c_B B^{-1} A_j. \tag{4}$$

Do $\overline{c}_j$ là sự thay đổi giá tạo nên khi ta thay đổi biến x_j với một lượng là đơn vị nên khi đưa các biến có giá âm vào cơ sở thì nghiệm cơ sở chấp nhận được mới sẽ cải thiện giá trị của hàm mục tiêu. Nếu không tồn tại biến với giá âm thì nghiệm có được chính là nghiệm tối ưu.

Mặc dù phương pháp đơn hình là phương pháp giải rất hiệu quả đối với nhiều bài toán thực tế nhưng việc định giá hiện như trên đòi hỏi phải duyệt toàn bộ các biến ở mỗi vòng lặp. Điều này trở nên không khả thi trong một số bài toán khác khi số lượng biến là rất lớn (bùng nổ tổ hợp) như trường hợp bài toán cắt vật tư. Vì vậy chúng ta cần có phương pháp xác định một cách hiệu quả biến có giá âm để thay thế các biến cơ sở. Kỹ thuật tạo sinh cột chính là kỹ thuật để khắc phục nhược điểm của phương pháp đơn hình khi giải các bài toán cỡ lớn. Phương pháp được thực hiện như sau.

Xuất phát từ ma trận cơ sở bất kỳ B ta tìm được một nghiệm cơ sở. Thay vì duyệt toàn bộ giá của các biến người ta chỉ xét đến các biến tương ứng với các cột có tiềm năng cải thiện giá trị hàm mục tiêu tại mỗi vòng lặp. Cột đó chính là vector nghiệm u^* của bài toán đối ngẫu (3) với ràng buộc được thay bởi ma trận cơ sở B:

$$w_{LP} = \max\{ub|uB \le c,\ u \in R_+^m\}. \tag{5}$$

Nếu ta thay A_j trong (4) bởi u^* và định giá ban đầu cho các cột được xác định bằng hàm $c(u)$ ta nhận được giá tương ứng với u^* như sau:

$$\overline{c}(u^*) = c(u^*) - c_B B^{-1} u^*. \tag{6}$$

Trong trường hợp $\overline{c}(u^*) \ge 0$, nghiệm cơ sở chính là nghiệm tối ưu. Ngược lại, ta thay cột bất kỳ trong ma trận B bởi u^* để nhận được ma trận cơ sở mới và tiếp tục lặp lại quá trình tính toán như trên với ma trận cơ sở vừa nhận được cho đến khi không còn tồn tại u^* để $\overline{c}(u^*) < 0$. Nghiệm cơ sở nhận được cuối cùng chính là nghiệm tối ưu của bài toán (2).

Mục sau sẽ dành để trình bày kỹ thuật tạo sinh cột để giải bài toán cắt vật tư một chiều 1DCSP.

2.2. Kỹ thuật tạo sinh cột để giải bài toán 1DCSP

Bài toán cắt vật tư được xác định bởi các dữ liệu sau $(m, L, l = (l_1, ..., l_m), b = (b_1, ..., b_m))$, trong đó L là bề rộng của tấm vật liệu thô, m là số kiểu vật liệu thành phẩm được cắt từ vật liệu thô và đối với mỗi kiểu vật liệu thành phẩm j, l_j là bề rộng và b_j là đơn hàng cho loại vật liệu thành phẩm đó. Bài toán đặt ra là tìm cách cắt sao cho số lượng tấm vật liệu thô sử dụng là ít nhất mà vẫn đáp ứng được đơn hàng.

Bài toán trên lần đầu tiên được Kantorovich [22] mô hình hóa vào năm 1939. Mô hình này có suy biến liên tục (continuous relaxation) rất yếu đồng thời việc giải mô hình này bắt buộc phải dùng phương pháp liệt kê.

Dantzig và Wolfe [28] đề xuất mô hình mới để khắc phục nhược điểm về suy biến liên tục yếu của mô hình trên, trong đó giới hạn mỗi vector $(x_{1j}, ..., x_{mj})$, $j = 1, ..., N$ nằm trong bao lồi của tất cả các phép cắt chấp nhận được. Mô hình này thường được gọi dưới thuật ngữ Dantzig-Wolfe Decomposition.

Gilmore và Gomory [20,21] lần đầu tiên đưa kỹ thuật tạo sinh cột (Column Generation) do Ford và Fulkerson đề xuất và sau đó được Dantzig và Wolfe hoàn thiện vào ứng dụng trong thực tế với việc giải bài toán cắt vật tư. Có thể tóm tắt phương pháp giải bài toán cắt vật tư một chiều của Gilmore và Gomory như sau.

Một phương án cắt được biểu diễn bởi một vector cột $a^j = (a_{1j}, ..., a_{mj}) \in Z_+^m$, $j = 1, ..., n$ sẽ cho biết số lượng vật tư thành phẩm được cắt từ tấm vật liệu thô. Phương án cắt được chấp nhận nếu nó thỏa mãn điều kiện:

$$\sum_{i=1}^{m} l_i a_{ij} \le L. \tag{7}$$

Giả sử x_j, $j = 1, ..., n$ là số lần phương án cắt chấp nhận được được sử dụng trong nghiệm. Mô hình của Gilmore và Gomory trở thành:

$$1DCSP_G(m, L, l, b) = \min \sum_{j=1}^{n} x_j = \min \|x\| \tag{8}$$

trên miền ràng buộc:

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_j,\ i = 1, ..., m, \tag{9}$$

$$x_j \in Z_+,\ j = 1, ..., n. \tag{10}$$

Mô hình (8)-(10) là bài toán quy hoạch nguyên với số lượng biến n tăng theo hàm lũy thừa của m. Mô hình này có suy biến liên tục rất mạnh với dự đoán về tính chất làm tròn nguyên cải biên (Modified Integer Round-Up Property MIRUP) như sau: *Sự sai khác giữa giá trị tối ưu của bài toán $1DCSP_G(m, L, l, b)$ và giá trị tối ưu của bài toán suy biến liên tục của nó luôn luôn nhỏ hơn 2.*

Trong thực tế người ta chưa chỉ ra được các ví dụ có sự sai khác lớn hơn 7/6 [9]. Hơn nữa những ví dụ có sai khác nhỏ hơn 1 chiếm đa số. Những bài toán như vậy được gọi là các bài toán có tính chất làm tròn nguyên (Integer Round-Up Property IRUP). Những bài toán có sai khác lớn hơn hoặc bằng 1 được gọi là những bài toán non-IRUP.

Trên cơ sở dự đoán đó, Gilmore và Gomory đề xuất cách tiếp cận giải bài toán (8)-(10) gồm 2 bước:

Bước 1: Giải bài toán suy biến liên tục của (8)-(10).

Bước 2: Làm tròn số nghiệm tối ưu của bài toán suy biến liên tục để nhận được nghiệm của bài toán (8)-(10).

Để giải bài toán suy biến liên tục của (8)-(10) với số lượng biến n rất lớn Gilmore và Gomory đề xuất sử dụng kỹ thuật tạo sinh cột (Column Generation), trong đó mỗi biến chỉ được sinh ra khi nó thực sự cần thiết cho việc tìm nghiệm. Sau khi thêm vào các biến bổ xung (slacks) ta có thể đưa bài toán (8)-(10) về dạng chuẩn:

$$1DCSP_G(m, L, l, b) = \min\{\|x\| : Ax = b,\ x \in Z_+^m\}. \tag{11}$$

Suy biến liên tục của (11) nhận được bằng việc loại bỏ ràng buộc nguyên trên các biến và được gọi là bài toán chủ (Master Problem - MP) sẽ có dạng:

$$1DCSP_G^{LP}(m, L, l, b) = \min\{\|x\| : Ax = b,\ x \in R_+^n\}. \tag{12}$$

Xuất phát, kỹ thuật tạo sinh cột sẽ làm việc với một tập con các cột của A được gọi là variable pool hoặc Restricted Master Problem (RMP). RMP có thể được khởi tạo dễ dàng, ví dụ bởi cơ sở kiến thiết ban đầu của phương pháp đơn hình. Giả sử A' là các cột trong A được lựa chọn. Khi đó RMP có dạng:

$$1DCSP_G^{LP}(m, L, l, b) = \min\{\|x\| : A'x = b,\ x \in R_+^n\}. \tag{13}$$

Do định giá ban đầu cho các biến là thuần nhất và bằng 1 nên giá suy giảm của một biến ứng với phương án cắt chấp nhận được a trong (13) được xác định bởi công thức $1 - ua$, với $u \in R_+^m$ là các giá trị biến đối ngẫu tối ưu của (13). Do vậy nghiệm tối ưu của bài toán xác định giá (Pricing Problem)

$$\max\{ua : al \leq L,\ a \in Z_+^m\} \tag{14}$$

sẽ lần lượt được thêm vào RMP nếu giá trị tối ưu của (14) lớn hơn 1. Nếu (14) không có nghiệm tối ưu như vậy thì nghiệm tối ưu của bài toán RMP (13) chính là nghiệm tối ưu của bài toán MP (12).

Trên cơ sở ý tưởng vừa trình bày, phương pháp tạo sinh cột để giải bài toán quy hoạch tuyến tính cỡ lớn MP có thể tóm tắt trong lược đồ sau:

```
Column Generation
    Begin
        Khởi tạo bài toán RMP
        While Tồn tại biến với giá suy giảm âm  Do
                Begin
                    Xác định biến có giá suy giảm âm nhỏ nhất
                    Bổ xung biến tìm được vào RMP
                    Giải lại RMP
                End
    End
```

Sau khi đã giải được bài toán $1DCSP_G^{LP}(m, L, l, b)$ bằng phương pháp tạo sinh cột, bước tiếp theo trong cách tiếp cận của Gilmore và Gomory là việc thực hiện thủ tục làn tròn số để nhận được nghiệm nguyên cho bài toán $1DCSP_G(m, L, l, b)$. Nhiều tác giả đã đưa ra các heuristics khác nhau để giải quyết vấn đề này. Một số thủ tục có thể tra cứu trong [8, 29].

Việc mở rộng bài toán cắt vật tư một chiều với nhiều kích thước vật liệu thô đã được một số tác giả đề cập đến [8, 9, 10, 16, 30]. Một nhận xét chung được đưa ra là khi cho phép có nhiều kích thước vật liệu thô, việc sử dụng vật liệu sẽ tốt hơn trường hợp chỉ có 1 loại vật tư. Tuy nhiên Holthaus [30] chỉ ra rằng việc xây dựng giải thuật heuristic cho bài toán là cần thiết vì các cách tiếp cận chính xác là không phù hợp trong trường hợp này. Điều này được chỉ ra một cách rõ ràng hơn trong [23], khi Rietz và Dempe đề xuất các nguyên tắc tạo nên các bài toán mà sự sai khác giữa nghiệm của bài toán suy biến liên tục và bài toán quy hoạch nguyên tương ứng trở nên rất lớn (tức là tính chất tương tự như IRUP đối với bài toán $1DCSP$ không được thỏa mãn). Các tác giả cũng đưa ra minh họa bằng các bài toán cụ thể được xây dựng từ những nguyên tắc đề ra.

Tiếp theo sẽ là một đề xuất lai ghép giữa thuật toán gen và cách tiếp cận của Gilmore và Gomory để giải lớp bài toán này.

3. PHƯƠNG PHÁP LAI GHÉP GIẢI BÀI TOÁN CẮT VẬT TƯ MỘT CHIỀU VỚI NHIỀU KÍCH THƯỚC VẬT LIỆU THÔ

3.1. Phát biểu bài toán

Bài toán cắt vật tư một chiều với nhiều kích thước vật liệu thô (One-Dimensional Cutting Stock Problem with Multiple Stock Sizes 1DMCSP) là mở rộng tự nhiên của bài toán 1DCSP trong đó các tấm vật liệu thô có thể có kích thước khác nhau. Bài toán 1DMCSP có thể được đặc trưng bằng các dữ liệu sau:

$$(m, M, l = (l_1, ..., l_m),\ b = (b_1, ..., b_m),\ L = (L_1, ..., L_M),\ c = (c_1, ..., c_M)), \tag{15}$$

trong đó m, l và b mang ý nghĩa như trong bài toán $1DCSP$; M là số các loại vật liệu thô sẽ được sử dụng và tham số L trong $1DCSP$ được thay thế bởi vector $L = (L_1, ..., L_M)$, với L_i là bề rộng và c_i là giá của vật liệu thô thứ i, $i = 1, ..., M$. Trong bài này ta giả thiết rằng giá của vật liệu thô tỷ lệ thuận với bề rộng của vật liệu.

Mô hình của bài toán trên (Machine Balance Model) được Gilmore và Gomory xây dựng trong [21]. Mô hình có thể được phát biểu như sau.

Đặt $m' = m + M$. Một vector $a = (a_1, ..., a_m) \in Z_+^{m'}$ là biểu diễn của một phương án cắt nếu

$$\sum_{i=1}^{m} l_i a_i \leq \sum_{i=1}^{M} L_i a_{i+m} \quad \text{và} \quad \sum_{i=1}^{M} a_{i+m} = 1. \tag{16}$$

Các thành phần a_i, $i = 1, ..., m$, xác định có bao nhiêu vật tư thành phẩm loại i được cắt. Thành phần a_{k+m} sẽ bằng 1 nếu loại vật liệu thô thứ k được sử dụng còn các thành phần a_{i+m} với $i \in \{1, ..., M\} \backslash k$ bằng 0.

Giả sử a_i, $i = 1, ..., n$ là tất cả các phương án cắt và mỗi thành phần x_i của vector $x = (x_1, ..., x_n)$ là số lần sử dụng phương án cắt a_i. Nhận xét rằng n có thể rất lớn. Ta xây dựng một vector n chiều c' từ c theo cách sau. Nếu phương án cắt a_i được thực hiện trên tấm thép loại k thì $c'_i = c_k$, $i = 1, ..., n$ và $k = 1, ..., M$. Khi đó ta có mô hình:

$$1DCSP(m, M, l, b, L, c) = \min c' x \tag{17}$$

trên miền ràng buộc:

$$\sum_{i=1}^{n} a_{ij} x_i \geq b_j,\ j = 1, ..., m, \tag{18}$$

$$\sum_{j=1}^{m} l_j a_{ij} \leq \sum_{j=1}^{M} L_j a_{i(j+m)}, \tag{19}$$

$$\sum_{j=1}^{M} a_{i(j+m)} = 1, \tag{20}$$

$$x \in Z_+^n, \tag{21}$$

$$a_i \in Z_+^{m'},\ i = 1, ..., n. \tag{22}$$

Từ các phát biểu (8)-(10) và (17)-(22) ta có mối liên hệ giữa hai bài toán như sau.

Định lý 1.

$$1DMCSP(m, M, l, b, L, c) \leq 1DCSP_G(m, L_k, l, b)\ \forall k \in \{1, ..., M\}.$$

Nói cách khác, bài toán cắt vật tư với nhiều kích thước vật liệu thô sẽ tốt hơn việc cắt chỉ trên một loại vật liệu.

Chứng minh. Do giả thiết giá của vật tư tỷ lệ thuận với bề rộng của nguyên liệu nên giá trị hàm mục tiêu chỉ phụ thuộc vào các phương án cắt. Các phương án cắt tối ưu cho (8)-(10) thỏa mãn (7) đối với chỉ số k nào đó thì cũng thỏa mãn (19) và do vậy chúng trở thành phương án chấp nhận được của (17)-(22). Điều này chứng minh khẳng định của định lý. ■

Định lý 1 khẳng định ý nghĩa của việc xây dựng các giải pháp cho bài toán cắt vật tư với nhiều kích thước vật liệu thô.

Giả sử x^* là nghiệm tối ưu của (17)-(22). Ta ký hiệu:

$\Omega(k),\ k = 1, ..., M$ là tập tất cả các phương án cắt trên vật liệu thô thứ k được sử dụng trong (17)-(22) tương ứng x^*;

$x^*/\Omega(k)$ là thu hẹp của x^* trên $\Omega(k)$;

b^k là vector vật tư thành phẩm nhận được từ (17)-(22) với việc sử dụng số lần cắt theo các phương án cắt trong $\Omega(k)$ được xác định bằng các thành phần tương ứng của $x^*/\Omega(k)$.

Định lý 2. *Nếu x^* là nghiệm tối ưu của bài toán $1DMCSP(m, M, l, b, L, c)$ (17)-(22) thì $x^*/\Omega(k)$ là nghiệm tối ưu của bài toán $1DCSP_G(m, L_k, l, b^k)$ $\forall k \in \{1, ..., M\}$ (8)-(10), $k = 1, ..., M$.*

Chứng minh. Giả sử x^* là nghiệm tối ưu của (17)-(22). Rõ ràng rằng nếu ta khởi tạo bài toán (8)-(10) với các phương án cắt được xác định trong $\Omega(k)$ thì $x^*/\Omega(k)$ chính là 1 nghiệm chấp nhận được của $1DCSP_G(m, L_k, l, b^k)$. Vì vậy ta có $1DCSP_G(m, L_k, l, b^k) \leq \|x^*/\Omega(k)\|$. Ta cần chỉ ra trong trường hợp này thực sự xảy ra dấu đẳng thức.

Giả sử $x^*/\Omega(k)$ không phải là nghiệm tối ưu của bài toán. Khi đó tồn tại x' là nghiệm của nó và thỏa mãn:

$$1DCSP_G(m, L_k, l, b^k) = \|x'\| < \|x^*/\Omega(k)\|.$$

Vì việc cắt nguyên liệu thô thứ k hoàn toàn độc lập với việc cắt các nguyên liệu thô khác nên nếu ta thay tập các biến của x^* nằm trong $x^*/\Omega(k)$ bằng tập các biến tương ứng của x' để nhận được tập biến mới x'' và đưa các phương án cắt tạo nên x' trong bài toán $1DCSP_G(m, L_k, l, b^k)$ vào (18) và (19) thì x'' chính là một nghiệm chấp nhận được của (17)-(22). Khi đó giá của hàm mục tiêu của bài toán (17)-(22) xác định trên x'' sẽ nhỏ hơn giá của hàm mục tiêu xác định trên x^*. Điều này mâu thuẫn với giả thiết x^* là nghiệm tối ưu của (17)-(22). Như vậy $x^*/\Omega(k)$ là nghiệm tối ưu của bài toán $1DCSP_G(m, L_k, l, b^k)$. Định lý được chứng minh. ■

Trên cơ sở của Định lý 2, ta có thể mô hình hóa bài toán $1DMCSP$ dưới dạng phát biểu mới như sau.

$$1DMCSP(m, M, l, b, L, c) = \min \sum_{k=1}^{M} \sum_{k=1}^{M} 1DCSP_G(m, L_k, l, b^k)c_k \tag{23}$$

trên miền ràng buộc:

$$\sum_{k=1}^{M} b^k \geq b, \tag{24}$$

$$b^k \in Z_+^m. \tag{25}$$

Định lý 3. *Nghiệm tối ưu của bài toán (23)-(25) sẽ xác định trên tập các vector b^k thỏa mãn $\sum_{k=1}^{M} b^k = b$. Tập các vector b^k như vậy được gọi là phân hoạch của b. Nói cách khác nghiệm tối ưu của (23)-(25) được xác định trên tập các phân hoạch của vector b.*

Chứng minh. Giả sử các b^k thỏa mãn $\sum_{k=1}^{M} b^k > b$. Khi đó ta có thể chọn các vector $t^k \in Z_+^m$ thỏa mãn $t^k \leq b^k$, $k = 1, ..., M$, $\sum_{k=1}^{M} t^k = b$. Rõ ràng rằng nếu x^* là nghiệm tối ưu của $1DCSP_G(m, L_k, l, b^k)$ thì x^* cũng là một nghiệm chấp nhận được của bài toán $1DCSP_G(m, L_k, l, t^k)$. Do đó $1DCSP_G(m, L_k, l, t^k) \leq 1DCSP_G(m, L_k, l, b^k)$, $k = 1, ..., M$. Thay các bất đẳng thức này vào (23), định lý được chứng minh. ■

Định lý 3 sẽ giới hạn không gian tìm kiếm của bài toán 1DMCSP trong tập các phân hoạch của vector đơn hàng.

3.2. Lai ghép thuật toán gen và kỹ thuật tạo sinh cột giải bài toán 1DMCSP

Các Định lý 2, 3 và phát biểu mới của bài toán $1DMCSP$ trong (23)-(25) gợi cho chúng ta ý tưởng phân rã bài toán $1DMCSP$ thành các bài toán cơ sở là bài toán $1DCSP_G(m, L_k, l, b^k)$ để có thể áp dụng được phương pháp giải sử dụng kỹ thuật tạo sinh cột của Gilmore và Gomory. Các nghiệm tối ưu của từng bài toán cơ sở sẽ được kết hợp lại để tạo nên nghiệm chấp nhận được của bài toán $1DMCSP$. Việc tìm nghiệm tối ưu cho bài toán $1DMCSP$ sẽ do thuật toán gen đảm nhiệm với không gian tìm kiếm là tập các phân hoạch của vector đơn hàng.

Sau đây ta sẽ lần lượt hình thức hóa ý tưởng đó trên ngôn ngữ của thuật toán gen và kỹ thuật tạo sinh cột.

Biểu diễn bài toán: Ta sử dụng nhiễn sắc thể có cấu trúc $(b^1, ..., b^M)$, $b^k \in Z_+^m$ để biểu diễn các cá thể (các điểm) trong không gian tìm kiếm. Mỗi quần thể là một tập bao gồm một số cố định các cá thể.

Độ đo thích nghi: Với mỗi cá thể $s = (b^1, ..., b^M)$ ta xác định mức độ thích nghi của cá thể $f(s)$ bằng công thức sau:

$$f(s) = \frac{1}{\sum_{k=1}^{M} 1DCSP_G(m, L_k, l, b^k)c_k}. \tag{26}$$

Toán tử hôn phối: Giả sử $s_1 = (b_1^1, ..., b_1^M)$ và $s_2 = (b_1^1, ..., b_2^M)$ là hai cá thể bất kỳ, k là một số được lựa chọn ngẫu nhiên, $1 < k \leq m$. Từ hai cá thể trên ta tạo ra hai hậu duệ s_1' và s_2' với các vector cột tương ứng của chúng được xác định như sau:

$$b_1'^j[i] = b_1^j[i],\ i = 1, ..., k-1,\quad b_1'^j[i] = b_2^j[i],\ i = k, ..., m,\ j = 1, ..., M, \tag{27}$$

$$b_2'^j[i] = b_2^j[i],\ i = 1, ..., k-1,\quad b_2'^j[i] = b_1^j[i],\ i = k, ..., m,\ j = 1, ..., M, \tag{28}$$

Toán tử đột biến: Chọn ngẫn nhiên một bộ 3 các số nguyên (p, q, r), $1 \leq p,\ q \leq M$ và $1 \leq r \leq m$, với xác suất $p_0 = 1/mM^2$. (29)

Toán tử đột biến tác động lên cá thể $s = (b^1, ..., b^M)$ để tạo nên cá thể $s = (b_1^1, ..., b_1^M)$ với bộ (p, q, r) đã chọn như sau:

$$b_1^i = b^i \text{ khi } i \neq p \text{ và } i \neq q,\ i = 1, ..., M, \tag{30}$$

$$b_1^p[j] = b^p[j],\ b_1^q[j] = b^q[j] \text{ khi } j = 1, ..., m \text{ và } j \neq r, \tag{31}$$

$$b_1^p[j] = b^p[r] - 1,\ b_1^q[r] = b^q[r] + 1 \text{ nếu } b^p[r] > 0, \tag{32}$$

$$b_1^p[r] = b^p[r],\ b_1^q[r] = b^q[r] \text{ nếu } b^p[r] = 0. \tag{33}$$

Toán tử chọn lọc: Toán tử chọn lọc được xác định theo luật tỷ lệ thuận với mức độ thích nghi:

$$p_s = \frac{f(s)}{\sum_{s \in G} f(s)} \tag{34}$$

trong đó, s là cá thể và G là quần thể đang xem xét có chứa s.

Định lý 4. *Giả sử hai cá thể cha-mẹ là các phân hoạch của cùng một vector. Khi đó các toán tử hôn phối và toán tử đột biến xác định như trên sẽ bảo đảm các hậu duệ cũng là những phân hoạch của vector đó.*

Chứng minh. Nếu ta biểu diễn các cá thể dưới dạng bảng có kích thước $m \times M$, trong đó cột i của bảng chứa vector b^i của cá thể, $i = 1, ..., M$. Khi đó toán tử hôn phối sẽ thực hiện việc trao đổi $M - k$ hàng cuối cùng của 2 bảng tương ứng với 2 phần tử cha-mẹ cho nhau để tạo nên 2 hậu duệ. Còn đối với toán tử đột biến, hậu duệ sẽ hoặc là bản sao của cha-mẹ theo (30)-(31)-(33), hoặc chỉ thay đổi tại hàng r theo (30)-(31)-(32). Vì vậy tổng các cột của hậu duệ luôn bằng tổng các cột của cha-mẹ. Định lý được chứng minh. ■

Dựa trên các Định lý 1,2,3,4 chúng ta có thể xây dựng thuật toán lai ghép như sau.

Thuật toán GA-CG

Input: m, M, l, L, b, c

Output: Nghiệm tối ưu của bài toán $1DMCSP(m, M, l, b, L, c)$ và các phương án cắt tương ứng với nghiệm tối ưu đó.

Step 0. Khởi tạo quần thể gồm K cá thể $G_0 = \{s_1^0, ..., s_K^0\}$. Việc khởi tạo này có thể thực hiện dễ dàng bằng việc tạo ra K phân hoạch khác nhau của b, mỗi phân hoạch sẽ tương ứng với một đối tượng của quần thể. Các cá thể được biểu diễn như sau:

$$s_i^0 = (b_{i1}^0, ..., b_{iM}^0),\ b_{ij}^0 \in Z_+^m,\ \sum_{j=1}^{M} b_{ij}^0 = b,\ i = 1, ..., K.$$

Step 1. Giải các bài toán $1DCSP_G(m, L_k, l, b_{ik}^t)$, $i = 1, ..., K$, $k = 1, ..., M$, t là thứ tự bước lặp (thứ tự của quần thể) bằng phương pháp tạo sinh cột. Tính mức độ thích nghi $f(s_i^t)$ cho từng cá thể của G_t theo (26).

Step 2. Lựa chọn các cha-mẹ trong G_t theo mức độ thích nghi để ghép cặp theo toán tử hôn phối (27)-(28) để tạo nên tập các hậu thế G_t' với K_1 phần tử.

Step 3. Tác động toán tử đột biến (30)-(33) vào $G_t \cup G_t'$ để nhận được G_t''.

Step 4. Thực hiện tính toán giống như trong Step1 cho các cá thể của G_t''.

Step 5. Áp dụng toán tử chọn lọc (34) lên $G_t \cup G_t''$ để chọn ra K cá thể có mức độ thích nghi lớn nhất tạo quần thể mới G_{t+1}.

Step 6. Nếu điều kiện dừng chưa thỏa mãn quay lại Step 2. Ngược lại thuật toán dừng và cho nghiệm tối ưu cùng tập các phương án cắt tương ứng với nghiệm tối ưu.

3.3. Tính hội tụ của thuật toán

Với giả thiết giá của vật liệu thô tỷ lệ thuận với bề rộng của vật liệu, ta có một số tính chất sau.

Bổ đề 1. *Nghiệm tối ưu của bài toán 1DMCSP nằm trong khoảng sau:*

$$\underline{O} = \min_k\{c_k\} \leq 1DMCSP(m, M, l, b, L, c) < \max_k\{c_k \sum_{j=1}^{m} b_j\} = \overline{O}. \tag{35}$$

Và xác suất chọn lọc trở thành cá thể thuộc quần thể tiếp theo của từng cá thể s thỏa mãn:

$$p_s \geq \frac{1/\overline{O}}{(2K+K_1)/\underline{O}} = p_1 > 0. \tag{36}$$

Chứng minh. Thật vậy, trong (35) bất đẳng thức đầu tiên xảy ra do có ít nhất một loại vật liệu thô được đưa vào sản xuất để đáp ứng đơn hàng. Đẳng thức sẽ đạt được trong trường hợp chỉ cần dùng 1 tấm vật liệu thô nhỏ nhất là đủ để sản xuất. Nói cách khác nếu ký hiệu $L_* = \min_k\{L_k\}$ thì đẳng thức xảy ra khi thỏa mãn:

$$L_* \geq \sum_{j=1}^{m} l_j b_j.$$

Bất đẳng thức thứ hai là hiển nhiên khi ta sử dụng mỗi vật liệu thô có bề rộng lớn nhất chỉ để cắt 1 vật tư thành phẩm.

Bất đẳng thức (36) được suy ra trực tiếp từ (34), (35) và cơ chế hoạt động của thuật toán GA-CG.

Ta ký hiệu:

$Partition(b)$: tập tất cả các phân hoạch của b thành M phân hoạch;

$F_t = \max\{f(s) : s \in G_t\}$: mức độ thích nghi lớn nhất của quần thể G_t;

$F^* = \max\{f(s) : s \in Partition(b)\}$: mức độ thích nghi lớn nhất của tất cả không gian nghiệm. Mức độ thích nghi lớn nhất này chính là nghịch đảo của nghiệm tối ưu toàn cục của bài toán $1DMCSP$;

$S^* = \max\{s \in Partition(b) : f(s) = F^*\}$: tập tất cả các cá thể tương ứng với nghiệm tối ưu toàn cục của bài toán $1DMCSP$.

Ta thấy, mỗi khi $F_t = F^*$ thì quần thể G_t sẽ chứa một cá thể biểu diễn nghiệm tối ưu toàn cục của bài toán $1DMCSP$.

Định nghĩa 1. Ký hiệu biến ngẫu nhiên $F = \min\{t \geq 0 : F_t = F^*\}$ là lần đầu tiên trong quần thể G_t chứa cá thể biểu diễn nghiệm tối ưu toàn cục của bài toán $1DMCSP$. Khi đó ta nói rằng thuật toán GA-CG sẽ đạt được nghiệm tối ưu toàn cục trong khoảng thời gian hữu hạn với xác suất 1 nếu $P\{F < \infty\} = 1$ không phụ thuộc vào quần thể khởi đầu.

Định nghĩa 2. Giả sử s và s' là hai cá thể khác nhau trong $Partition(b)$. Cá thể s' được gọi là có thể đạt được từ s nếu tồn tại một dãy hữu hạn các cá thể $s_1, ..., s_r$ đôi một khác nhau của $Partition(b)$ sao cho $s = s_1,\ s' = s_r$ và s_{i+1} nhận được từ s_i với xác suất dương bằng việc áp dụng toán tử đột biến vào $s_i,\ i = 2, ..., r$.

Bổ đề 2. *Mọi cá thể trong Partition(b) đều có thể đạt được từ cá thể bất kỳ của Partition(b).*

Chứng minh. Giả sử s, s' là hai cá thể khác nhau bất kỳ trong $Partition(b)$. Nếu ta biểu diễn hai cá thể đó bằng hai ma trận A và A' kích thước $m \times M$, ta có thể định nghĩa khoảng cách giữa hai cá thể này là:

$$d(s, s') = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{M}|a_{ij} - a'_{ij}|.$$

Do s và s' là các phân hoạch khác nhau của vector b nên nếu tại một dòng r nào đó của ma trận A tồn tại một chỉ số p sao cho $a_{rp} > a'_{rp}$ thì cũng sẽ tồn tại một chỉ số q mà tại đó thỏa mãn $a_{rp} < a'_{rp}$. Khi đó ta có thể thực hiện phép đột biến trên bộ 3 (p, q, r) với xác suất p_0 vào s để nhận được cá thể mới s'' cũng là phần tử của $Partition(b)$. Khi đó $d(s'', s') = d(s, s') - 1$. Thực hiện lên tiếp thủ tục này $d(s, s')$ lần ta sẽ nhận được s'.

Theo giả thiết ban đầu, s và s' là hai cá thể bất kỳ nên bổ đề được chứng minh. ■

Định lý 5. *Thuật toán GA-CG đạt được nghiệm tối ưu toán cục trong khoảng thời gian hữu hạn với xác suất 1 không phụ thuộc vào quần thể khởi đầu.*

Chứng minh. Giả sử s, s' là hai cá thể của $Partition(b)$ sao cho $s \notin S^*$ và $s' \notin S^*$. Theo Bổ đề 2, s' có thể đạt được từ s, tức là tồn tại một nhánh hữu hạn đi từ s đến s' mà mỗi cá thể trên đó nhận được bằng việc tác động toán tử đột biến vào cá thể đứng trước nó. Dễ dàng chỉ ra rằng độ dài của nhánh ngắn nhất đi từ s tới s' bằng các toán tử đột biến chính là $d(s, s')$. Đặt $k_s = \max\{d(s, s') : s' \in S^*\}$ và $k = \max\{k_s : s \notin S^*\}$.

Xét một cá thể tùy ý s của một quần thể nào đó. Theo thuật toán GA-CG, Step 3 bảo đảm rằng cá thể này chắc chắn được tác động bởi toán tử đột biến và xác suất để ảnh của nó qua toán tử đột biến trở thành điểm tiếp theo trên đường đi ngắn nhất từ s đến S^* sẽ không nhỏ hơn p_0. Theo (34) và ước lượng (36), ta có xác suất để một cá thể bất kỳ được lựa chọn làm cá thể thuộc quần thể thế hệ sau không nhỏ hơn p_1. Như vậy xác suất để ảnh của s qua phép đột biến đồng thời là điểm tiếp theo trên đường đi ngắn nhất từ s tới S^* và được lựa chọn làm cá thể của quần thể thế hệ sau sẽ không nhỏ hơn $p_0^k p_1^{k_s-1}$. Lặp liên tiếp k_s lần lập luận như trên, ta thấy xác suất chuyển từ $s \notin S^*$ vào thành cá thể nào đó thuộc S^* tại bước lặp thứ k_s sẽ không nhỏ hơn $p_0^{k_s} p_1^{k-1} > 0$. Vì vậy ta có thể đánh giá xác suất để một cá thể tối ưu toàn cục được xét đến trong thuật toán sau k vòng lặp sẽ không nhỏ hơn $p_0^k p_1^{k-1} > 0$ và không phụ thuộc vào sự khởi đầu thực sự của $s \notin S^*$. Từ đây suy ra rằng, xác suất để thuật toán không xét đến bất kỳ một nghiệm tối ưu nào sau t vòng lặp sẽ lớn nhất là

$$(1 - p_0^k p_1^{k-1})^{\lfloor t/k \rfloor}. \tag{37}$$

Dễ thấy rằng đại lượng trong (37) sẽ hội tụ rất nhanh theo tốc độ lũy thừa tới 0 khi $t \to \infty$. Điều này có nghĩa $P(F < \infty) = 1$ (hay là nghiệm tối ưu toàn cục được thuật toán xét đến lần đầu tiên sau hữu hạn bước lặp với xác suất bằng 1). Do s là cá thể tùy ý nên định lý được chứng minh. ■

4. NGHIÊN CỨU SO SÁNH

Trong các công trình liên quan đến bài toán cắt vật tư 1 chiều, hầu hết các tác giả đều có chung dự đoán là bài toán sẽ nghiệm tối ưu tốt hơn khi sử dụng nhiều loại vật liệu thô [8,9,10,16,20,21]. Tuy nhiên dự đoán đó chỉ dừng lại ở việc đưa ra nhận xét. Trong bài báo này, Định lý 1 đã góp phần khẳng định dự báo đó là đúng.

Xuất phát từ nhận xét trong [30], các tác giả đều có khuynh hướng lai ghép một thủ tục heuristic với các thuật toán giải bài toán cắt vật tư 1 chiều cho một loại vật liệu thô như các thuật toán Branch-And-Bound, Branch-And-Price, Cutting Plane ... để áp dụng giải bài toán cắt vật tư 1 chiều cho nhiều loại vật liệu thô. Các đề xuất chỉ bao gồm việc phát biểu mô hình bài toán và mô tả các bước của thuật toán lai ghép mà không chứng minh tính đúng đắn của thuật toán đưa ra (bảo đảm tìm được nghiệm tối ưu sau khoảng thời gian hữu hạn). Hầu hết các cách tiếp cận này đều dẫn đến bùng nổ tổ hợp đòi hỏi thời gian tính toán quá lớn nên ít có khả năng triển khai thực tiễn. Chính vì vậy Belov, một trong những tác giả quan tâm nhiều tới vấn đề này và cũng là một thành viên của ESICUP (Euro Special Interest Group on Cutting and Packing - http://paginas.fe.up.pt/ esicup/tiki-index.php?page=About%20Esicup), đã nhận xét trong các công trình của mình là cho đến nay chưa có một công trình nào về hiệu quả thực tiễn của các giải pháp cho bài toán cắt vật tư 1 chiều với nhiều loại vật liệu thô được công bố [8,9,10,16,25].

Thuật toán được đề xuất trong bài này bảo đảm hội tụ hầu chắc chắn đến nghiệm tối ưu (Định lý 5). Điều đó khẳng định độ tin cậy của thuật toán được đưa ra so với các thuật toán của các tác giả khác.

Cũng trong bài báo này, mối liên hệ ngữ nghĩa của bài toán cắt vật tư 1 chiều cho nhiều loại vật tư với bài toán cắt vật tư 1 chiều cho một loại vật tư được thể hiện trong các Định lý 1,2,3. Từ đó, ta có thể song song và phân tán hóa việc cài đặt thuật toán để tăng tính hiệu quả về thời gian. Trong thực tế cài đặt, nhóm tác giả đã xây dựng thuật toán dưới dạng một hệ thống đa tác tử (Multi-agent System) gồm 2 lớp tác tử. Lớp trên gồm 1 agent thực hiện thuật toán gen có nhiệm vụ lập kế hoạch (tạo ra các phân hoạch của vector kế hoạch), điều phối các agent ở lớp dưới và đồng bộ hóa kết quả (tính độ đo thích nghi, thực hiện các toán tử gen...). Lớp dưới gồm các agent thuần nhất (các agent đều cùng thực hiện thuật toán tạo sinh cột để giải bài toán cắt vật tư 1 chiều cho 1 loại vật liệu thô). Các agent của lớp dưới là các agent di động (Mobile agent) được agent lớp trên tạo sinh khi có nhu cầu và có thể khu trú tại những node tính toán bất kỳ trên mạng cục bộ có đủ tài nguyên. Đây cũng là điều vượt trội của thuật toán của chúng tôi so với các thuật toán mà các tác giả khác đề xuất. Việc cài đặt như vậy đã tận dụng được sức mạnh tính toán của mạng LAN và góp phần giảm đáng kể thời gian giải bài toán (vấn đề này chúng tôi sẽ có một bài báo tiếp theo để chi tiết hóa việc cài đặt hệ thống đa tác tử). Hệ thống này đã được triển khai phục vụ sản xuất tại nhà máy ống thép Việt-Đức trong 4 năm qua. Kết quả cho thấy trên thực tế chưa gặp một đợt sản xuất nào có lượng phế thải lớn hơn 3% bề rộng của tấm vật tư có kích thước lớn nhất. Điều đó chỉ ra rằng thuật toán GA-CG được thiết kế như trên có tính chất tương tự như tính chất IRUP bằng chính các thẩm định thực tế, điều mà các tác giả khác không đề cập tới.

5. KẾT LUẬN

Bài báo này đưa ra một phân tích tổng quan về thực trạng nghiên cứu bài toán cắt vật tư một chiều trong những năm gần đây và trình bày tóm tắt kỹ thuật tạo sinh cột của Gilmore và Gomory để giải bài toán này với trường hợp có 1 loại vật liệu thô. Thuật toán này là thuật toán được sử dụng như một mặc định khi nói đến cắt vật tư vì tính hiệu quả của nó.

Bài toán cắt vật tư 1 chiều với nhiều loại vật liệu thô được các nhà nghiên cứu thống nhất là chỉ có thể giải hiệu quả bằng các phương pháp Heuristics. Để có thể tận dụng sức mạnh của kỹ thuật tạo sinh cột, trong bài báo này chúng tôi đã đưa ra một cách phát biểu mới cho bài toán, chứng minh một số tính chất của cách phát biểu mới đồng thời đề xuất lai ghép thuật toán gen với kỹ thuật tạo sinh cột để tạo nên một Heuristic được đặt tên là GA-CG. Tính đúng đắn của GA-CG được chứng minh. Thuật toán đã được cài đặt và chạy thử nghiệm thực tế. Các kết quả thử nghiệm đã xác thực tính đúng đắn của các kết quả lý thuyết đồng thời cũng cung cấp nhiều dữ liệu làm minh chứng về tính chất IRUP của GA-CG.

TÀI LIỆU THAM KHẢO

[1] Phan Thị Hoài Phương, Nguyễn Minh Hằng, Mô hình kết hợp thuật toán gen và phương pháp đơn hình ứng dụng trong bài toán cực tiểu hóa chi phí sản xuất, *Tạp chí Tin học và Điều khiển học* **22** (4) (2006) 319–324.

[2] H. Dyckhoff, G. Wascher, (Eds.), Cutting and packing, *European Journal of Operational Research* **44** (2) (1990).

[3] P.Y. Wang, G. Wascher, (Eds.), Cutting and packing, *European Journal of Operational Research* **141** (2) (2002).

[4] G. Waăscher, H. Haussner, H. Schumann, An improved typology of cutting and packing problems, *European Journal of Operational Research* **183** (2007) 1109–1130.

[5] Jose Fernando Oliveira, Gerhard Waăscher, Cutting and packing, *European Journal of Operational Research* **183** (2007) 1106–1108.

[6] Shahin, A. Adham, Salem, M. Ossama, Using genetic algorithms in solving the one-dimensional cutting stock problem in the construction industry, *Canadian Journal of Civil Engineering* **31** (2) (2004) 321–332.

[7] Abbas Afshar, et. al., An improved linear programming model for one-dimensional cutting stock problem, *First International Conference on Construction In Developing Countries*, Karachi, 2008.

[8] G. Belov and G. Scheithauer, A cutting plane algorithm for the one-dimensional cutting stock problem with multiple stock lengths, *European Journal of Operational Research (Special issue on cutting and packing)* **141** (2) (2002) 274–294.

[9] G. Belov and G. Scheithauer, "A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting", Technical report, Dresden University, 2003, URL: www.math.tu-dresden.de/∼capad.

[10] G. Belov and G. Scheithauer, "Setup and open stacks minimization in one-dimensional stock cutting", Technical report, Dresden University, 2003.

[11] Silas Sallaume, et. al., "One dimensional cutting stock problem with redevelopment of the surplus material", Rio de Janeiro, Brazil, 1 - 5, June 2008.

[12] Sirirat Wongprakornkul, et. al., Optimization based heuristic approaches for solving an integrated one-dimensional cutting stock-transportation problem, *Journal of Mathematics and Statistics* **3** (3) (2007) 142–150.

[13] E. G. Coffman, M. R. Garey, and D. S. Johnson, Approximation algorithms for bin packing: A survey. *Approximation Algorithms for NP-Hard Problems*, in D. S. Hochbaum, editor, PWS Publishing Company, Boston, 1997.

[14] E. Falkenauer, Tapping the full power of genetic algorithm through suitable representation and local optimization. *Evolutionary Algorithms in Management Applications*, in J. Biethann and V. Nissen, editors, Springer-Verlag, Berlin, 1995.

[15] E. Falkenauer, A hybrid grouping genetic algorithm for bin packing, *Journal of Heuristics* **2** (1996) 5–30.

[16] Gleb Belov, Problems, Models and algorithms in one- and two-dimensional cutting, *Dissertation*, TU Dresden, 2004.

[17] Jacob Puchinger, Combining metaheuristics and integer programming for solving cutting and packing problem. Dissertation. Vienna University of Technology. http://www.ads.tuwien.ac.at/publications/bib/pdf/puchinger_06.pdf.

[18] G. Scheithauer and J. Terno, A branch& bound algorithm for solving one-dimensional cutting sock problems exactly, *Applicationes Matematicae* **23** (2) (1995) 151–167.

[19] Claudio Manuel Martins Alves, "Cutting and packing: problems, models and exact algorithms. Dissertation", Universidade de Minho. 2005.

[20] P. C. Gilmore, R. E. Gomory, A linear programming approach to the cutting-stock problem, *Oper. Res.* **9** (1961) 849–859.

[21] P. C. Gilmore, R. E. Gomory, A linear programming approach to the cutting stock problemPart II, *Oper. Res.* **11** (1963) 863–888.

[22] L. V. Kantorovich, Mathematical methods of organizing and planning production, *Management Science* **6** (1960) 363–422 (in Russian 1939).

[23] J. Rietz and S. Dempe, Large gaps in one-dimensional cutting stock problems, *Discrete Applied Mathematics* **156** (10) (2008).

[24] Antonio Carlos Moretti, et. al., Nonlinear cutting stock problem model to minimize the number of different patterns and objects, *Computational & Applied Mathematics* **27** (1) (2008) 61–78.

[25] Greb Belov and Adam N. Letchford, A node-flow model for 1D stock cutting: Robust branch-cut-and-price. 2005, http://www.producao.uff.br/conteudo/rpep/volume52005/RelPesq_V5_2005_07.pdf.

[26] Hildegard Foerster, Gerhard Wascher, Pattern reduction in one-dimensional cutting stock problems, *Journal of Production Research* **38** (7) (2000) 1657–1676.

[27] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, New York, 1979.

[28] G.B. Dantzig, P. Wolfe, Decomposition Principle for Linear programs, *Operations Research* **8** (1960) 101–111.

[29] Sirinat Wongprakornkul, Round down technique for solving an integer linear programming, *KKU Science Journal* **36** (2008) 187–198.

[30] O. Holthaus, Decomposition approaches for solving the integer one dimensional cutting stock problem with different types of standard lengths, *European Journal of Operational Research* **141** (2002) 295–312.

Nhận bài ngày 5 - 3 - 2009
Nhận lại sau sửa ngày 3 - 11 - 2009