

VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



MATHEMATICAL MODELING (CO2011)

Assignment

“2D Cutting Stocks Problem”

Instructor(s): Mai Xuân Toàn

Students: Lữ Quốc Pháp - 2312562 (*Leader*)
Võ Thị Ánh Tuyết - 2213826
Hồ Quang Vinh - 2213960
Võ Huỳnh Anh Thư - 2213416

HO CHI MINH CITY, DECEMBER 2022



Contents

List of Symbols	2
List of Acronyms	2
List of Figures	4
List of Tables	4
Member list & Workload	4
1 Phân tích thuật toán cho tool của nhóm	5
1.1 Genetic Algorithm	5
1.2 First Fit Decreasing	6



List of Symbols

\mathbb{N} Set of natural numbers

\mathbb{R} Set of real numbers

\mathbb{R}^+ Set of positive real numbers

List of Acronyms

ODE (First-Order) Ordinary Differential Equation

IVP Initial-Value Problem

LTE Local Truncation Error

DS Dynamical System

Fig. Figure

Tab. Table

Sys. System of Equations

Eq. Equation

e.g. For Example

i.e. That Is



List of Figures

1	Quan hệ giữa các phương thức	6
---	--	---

List of Tables

1	Member list & workload	4
---	----------------------------------	---



Member list & Workload

No.	Fullname	Student ID	Problems	% done
1	Lữ Quốc Pháp	2312562	- Exercise 1: 1.2 - Exercise 2 - Exercise 3: 3.2	100%
2	Võ Thị Ánh Tuyết	2213826	- Exercise 1: 1.3 - Exercise 2 - Exercise 3: 3.1 - Exercise 4	100%
3	Hồ Quang Vinh	2213960	- Exercise 1: 1.1 - Exercise 4 - L ^A T _E X	100%
4	Võ Huỳnh Anh Thư	2213416	- Exercise 1 - Exercise 3: 3.1 - Exercise 4	100%

Table 1: Member list & workload

1 Phân tích thuật toán cho tool của nhóm

1.1 Genetic Algorithm

Trong khoa học máy tính, giải thuật di truyền là một metaheuristic được lấy cảm hứng từ quá trình chọn lọc tự nhiên. Thuật toán này được sử dụng để tìm ra các giải pháp tối ưu cho các bài toán tìm kiếm và tối ưu hóa, thông qua các khái niệm sinh học như *selection* (chọn lọc), *crossover* (lai ghép), và *mutation* (đột biến).

Trong thuật toán di truyền, quần thể là tập hợp các giải pháp khả thi cho một bài toán tối ưu hóa và được tiến hóa để trở nên tốt hơn. Mỗi giải pháp (hay cá thể) có một bộ thuộc tính (gọi là nhiễm sắc thể), có thể bị đột biến hoặc thay đổi trong quá trình tiến hóa.

Giải thuật bắt đầu với một quần thể ban đầu mà các cá thể được sinh ngẫu nhiên dựa trên cách thể hiện của một cá thể. Trong mỗi thế hệ, độ thích nghi (fitness) của từng cá thể được đánh giá (độ thích nghi thường là giá trị của hàm mục tiêu). Những cá thể có độ thích nghi cao sẽ được chọn từ thế hệ hiện tại và các gen của chúng sẽ được biến đổi (bằng cách lai ghép hoặc đột biến) để tạo ra thế hệ mới. Thế hệ mới này sau đó sẽ được sử dụng trong *iteration* (lặp) tiếp theo của thuật toán.

Thuật toán sẽ dừng lại khi số lượng thế hệ đã đạt tới mức tối đa hoặc khi đã đạt được độ thích nghi thỏa mãn yêu cầu của bài toán.

Trong thuật toán di truyền ta cần, chú ý những thuật ngữ quan trọng sau đây:

- **crossover** (lai hóa) đây là bước quan trọng để quyết định những cá thể với bộ **gene** nào sẽ được chọn lọc cho những thế hệ sau.
- **mutation** (đột biến) đây là một trường hợp riêng biệt trong quá trình lai hóa để tạo ra thế hệ mới. Đây là bước quan trọng trong việc tạo ra một bộ giải pháp mới từ những giải pháp cũ. Tưởng tượng quần thể không có **gene** nào được tạo ngẫu nhiên mà có ưu thế mạnh trong quần thể (thỏa điều kiện yêu cầu), cộng với việc không có đột biến xảy ra, thì quần thể đó sẽ không bao giờ được cải tiến. Nên đột biến là bước rất quan trọng để cải thiện sự đa dạng cho quần thể.

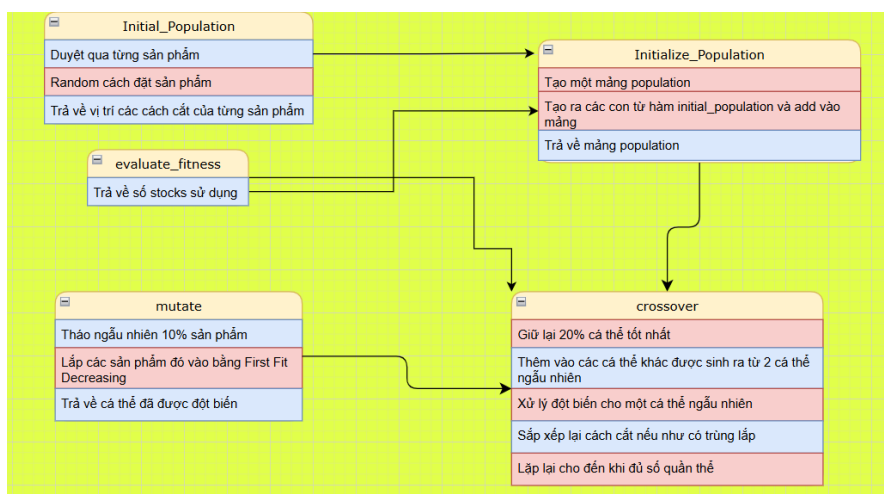


Figure 1: Quan hệ giữa các phương thức

Biểu đồ trên mô tả các bước hiện thực của các phương thức trong policy thuật toán di truyền. Mũi tên đi từ phương thức A đến phương thức B nghĩa là phương thức A sẽ được sử dụng bên trong phương thức B. Đây cũng là mã giả được nhóm sử dụng để hiện thực tool.

Về ưu điểm:

- Thuật toán có thể sử dụng trong những bài toán tối ưu lớn trong thời gian dài. Đối với bài toán 2D **Cutting Stock Problem**, chúng ta cần làm việc với rất nhiều biến trong hàm mục tiêu và các ràng buộc của bài toán. Vì vậy, giải thuật di truyền sẽ được sử dụng rất hiệu quả trong bài toán này.

Về nhược điểm:

- Thuật toán Genetic phụ thuộc khá nhiều vào các thông số như **Population Size** (khối lượng quần thể) hay **Generation** (số thế hệ). Vì vậy, chất lượng bài toán có thể giảm đi đáng kể nếu các thông số này không được khởi tạo một cách đúng đắn.
- Đây là giải thuật khá mạnh để giải quyết các bài toán độ khó **NP Hard**. Tuy nhiên, với các bài toán cần quan tâm đến nhiều thông số để tính toán như 2D cutting stocks trong trường hợp này, giải thuật này cần một lượng thời gian tính toán rất lâu. Điều này làm cho giải thuật trở nên thiếu thực tế.

1.2 First Fit Decreasing

Giải thuật **First Fit Decreasing (FFD)** là một giải thuật heuristic dùng để giải quyết các bài toán bin packing hoặc các bài toán liên quan đến tối ưu về mặt không gian. Đây là thuật toán cải tiến so với **First Fit**. Thuật toán FFD có cải tiến ở chỗ sắp xếp các sản phẩm vào kho theo thứ tự giảm dần kích thước.

Cụ thể, thuật toán sẽ từ từ lấy ra những sản phẩm bên trong danh sách các sản phẩm cần lắp. Và sẽ dò tìm các stocks phù hợp để lắp sản phẩm vào. Vị trí đầu tiên ứng với stock đầu tiên được cho là hợp lệ để gắn sản phẩm hiện tại vào sẽ là stock và vị trí được chọn để lắp sản phẩm vào.

Như vậy, ý đồ của thuật toán là dò tìm theo một thứ tự các stocks cụ thể và theo thứ tự đó bất cứ trường hợp nào có thể lắp sản phẩm hiện tại vào được thì trường hợp đó sẽ là một cách đặt chính thức trong danh sách các cách đặt sản phẩm.

Thuật toán này đảm bảo rằng một stock sẽ chứa được nhiều sản phẩm nhất có thể rồi mới đến những stocks sau. Vì vậy đây là một thuật toán tối ưu cục bộ.

Các bước quan trọng của thuật toán:

- Các sản phẩm cần lắp sẽ được sắp xếp lại theo thứ tự giảm dần kích thước để sử dụng cho việc duyệt danh sách sản phẩm đặt vào stocks. Sau đó, các sản phẩm sẽ được duyệt qua và lấy các thông tin về kích thước (size) và số lượng (quantity).
- Trong vòng lặp duyệt qua các sản phẩm. Khi lấy được thông tin về sản phẩm hiện tại, ta duyệt qua các stocks hiện có và lấy thông tin stock hiện tại. Duyệt qua từng vị trí trong stock đó, nếu có thể lắp được thì ta sẽ đánh dấu tại vị trí đó và giảm số lượng của prod đã lắp đi.
- Như vậy sau vòng lặp, các stocks có sản phẩm được lắp sẽ có độ lấp đầy cao và gần như không có stocks nào có tỉ lệ lấp đầy thấp. Vì vậy, Nếu các stocks đầu tiên được duyệt đều có kích thước nhỏ hơn nhiều so với các stocks nằm phía sau thì thuật toán sẽ không quá tối ưu. Đây là một nhược điểm của đặc tính tối ưu cục bộ.

Data: Danh sách `stocks` (các kho hàng) và `products` (các sản phẩm cần cắt)
Result: Hành động bao gồm chỉ số kho hàng (`stock_idx`), kích thước sản phẩm (`size`), và vị trí (`position`)

Bước 1: Sắp xếp danh sách `products` theo diện tích (`width × height`) giảm dần.

foreach `product` trong danh sách đã sắp xếp **do**

if `product.quantity == 0` **then**

 Bỏ qua sản phẩm này;

continue;

end

 Lấy kích thước của sản phẩm: `prod_w, prod_h = product.size`;

foreach `stock` trong danh sách `stocks` (chỉ số `stock_idx`) **do**

Bước 2: Lấy kích thước của kho hàng: `stock_w, stock_h`;

Bước 3: Kiểm tra nếu `stock` đủ lớn để chứa sản phẩm (không xoay hoặc có xoay):

if `stock_w ≥ prod_w` và `stock_h ≥ prod_h` **then**

foreach `(x, y)` là vị trí khả thi trong `stock` **do**

if `product` có thể đặt tại `(x, y)` **then**

 Trả về: `stock_idx, size, (x, y)`;

end

end

end

if `stock_w ≥ prod_h` và `stock_h ≥ prod_w` **then**

foreach `(x, y)` là vị trí khả thi trong `stock` **do**

if `product` có thể đặt tại `(x, y)` (sau khi xoay) **then**

 Trả về: `stock_idx, size[::-1], (x, y)`;

end

end

end

end

end

Bước 4: Nếu không tìm thấy vị trí phù hợp, trả về: `stock_idx = -1, size = [0, 0], position = (0, 0)`;

Algorithm 1: First Fit Decreasing Policy Pseudocode

Về ưu điểm:

- Đây là thuật toán tối ưu rất tốt về mặt thời gian so với các thuật toán tối ưu toàn cục. Thuật toán này thực hiện mang lại hiệu quả rất tốt cho các trường hợp có rất nhiều sản phẩm. Vì quá trình tính toán hành động diễn ra nhanh hơn nhiều so với các thuật toán tối ưu toàn cục mà vẫn mang lại kết quả khá tốt.

Về nhược điểm:

- Nhược điểm của thuật toán là không có độ tối ưu tuyệt đối. Đôi khi vẫn sinh ra sự lãng phí do không chủ động trong việc chọn stock sao cho phù hợp nhất với sản phẩm hiện tại.

References

- [1] P Wattanasiriseth and A Krairit, “ An Application of Cutting-Stock Problem in Green Manufacturing: A Case Study of Wooden Pallet Industry”, 2019.
- [2] Y.-H. Huang, H.-C. Lu, Y.-C. Wang, Y.-F. Chang, and C.-K. Gao, “A Thousand Ways to Pack the Bin- A Practical Approach to Two-Dimensional Rectangle Bin Packing”, 2010.
- [3] Bambang Santoso and Heri Haerudin, “Genetic Algorithm Approach for Cutting Stock Problems in Construction Industries” , 2023.
- [4] Robert W. Haessler and Paul E. Sweeney, “Cutting stock problems and solution procedures” *School of Business Administration, The University of Michigan, Ann Arbor, MI, USA*, 1991.
- [5] R. W. Haessler and P. E. Sweeney, “Cutting stock problems and solution procedures,” *European Journal of Operational Research*, vol. 54, no. 2, pp. 141–150, 1991.
- [6] Hari Mohan Pandey, “Performance Evaluation of Selection Methods of Genetic Algorithm and Network Security Concerns”, <https://doi.org/10.1016/j.procs.2016.02.004>, 2016.
- [7] Chassiakos AP, Rempis G., “Evolutionary algorithm performance evaluation in project time-cost optimization. *J Soft Comput Civ Eng* 2019;3(2):16–29”, <https://doi.org/10.22115/scce.2019.155434.1091>.