TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

Trường CNTT và Truyền thông

----- -----

Mini-Project

Đề tài: Sắp xếp kiện hàng container

GVHD: Bùi Quốc Trung

Nhóm 6

Danh sách thành viên:

Nguyễn Xuân Cường	**20190040**
Nguyễn Văn Chiến	**20183488**
Nguyễn Văn Sáng	**20194153**
Đoàn Tuấn Vũ	**20183672**

1.Mô tả bài toán

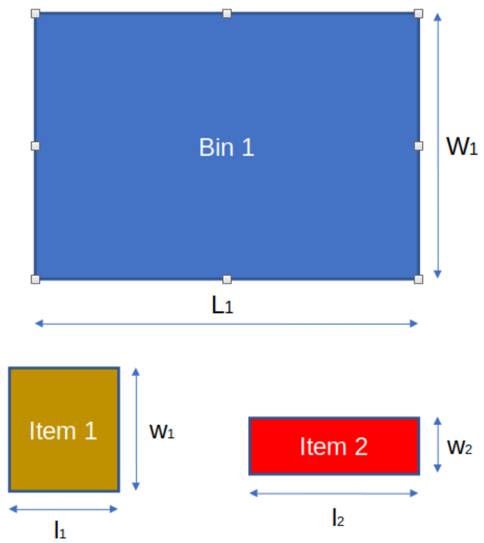
Có K xe tải(1,2,…K) vận chuyển N gói hàng(1,2,…N):
Trong đó:
Mỗi xe tải k:

+ kích thước Wk*Lk
+ chi phí sử dụng Ck

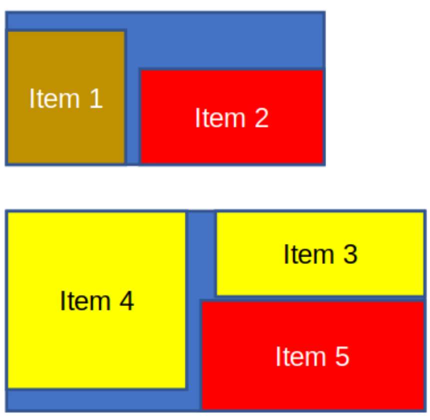
Mỗi gói hàng i:

+ kích thước wi x li
+ điều kiện các gói hàng không được chồng lên nhau

Số lượng xe K có thể lớn dẫn đến nhiều xe không dùng tới
Bài toán yêu cầu sắp xếp N gói hàng vào K xe sao cho tổng chi phí sử dụng các xe là nhỏ nhất

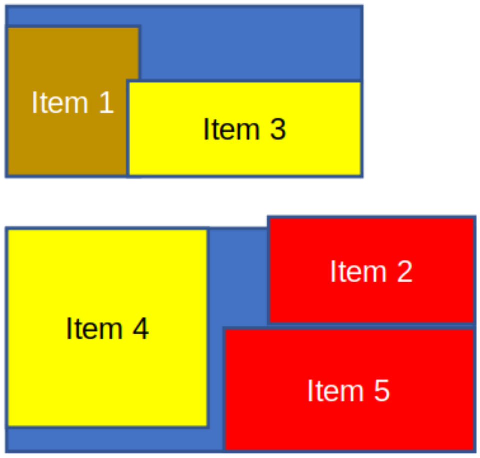


Sau đây là ví dụ trường hợp xếp các kiện hàng vào container hợp lệ và không hợp lệ



Hình 1: Trường hợp hợp lệ

Hình biểu diễn mô tả một trường hợp hợp lệ: Các kiện hàng sắp xếp trong thùng chứa không bị xếp chồng lên nhau cũng như các kiện hàng nằm gọn trong thùng hàng.



Hình 2: Trường hợp không hợp lệ

Một trường hợp được coi là không hợp lệ nếu như tồn tại thùng chứa có chứa hai kiện hàng xếp bị đè lên nhau hoặc có một kiện hàng không được xếp hoàn toàn ở phía trong của thùng

2. Phương pháp

a. Mixed-interger-programming

Các biến sử dụng:

Biến	Ý nghĩa	Ràng buộc
x_i, y_i	Toạ độ điểm góc dưới bên trái của gói hàng i	$0 \le x_i \le \max W_j, j \in (1, N)$ $0 \le y_i \le \max L_j, j \in (1, N)$
l_{ij}	Gói hàng i xếp bên trái gói j	$l_{ij} = 1$ or $l_{ij} = 0 \quad \forall i, j \in 1, N$
b_{ij}	Gói hàng i xếp bên dưới gói j	$b_{ij} = 1$ or $b_{ij} = 0 \; \forall i, j \in 1, N$
z_k	Thùng hàng xe thứ k được dùng	$z_k = 1$ or $z_k = 1 \quad \forall k \in 1, K$
f_{ik}	Gói hàng i được xếp trong xe k	$f_{ik} = 1$ or $f_{ik} = 0 \; \forall i \in 1, N; \forall k \in 1, K$

Các ràng buộc:

Ta có: $$W = \max_{j \in 1,N} W_j \qquad L = \max_{j \in 1,N} L_j$$

1. No overlap

$$l_{ij} + l_{ji} + b_{ij} + b_{ji} + (1 - \mathrm{f}_{ik}) + (1 - \mathrm{f}_{jk}) \geq 1, \forall i, j \in 1, N; k \in 1, K$$

2. Liên hệ toạ độ của 2 item i và j

$$y_i - y_j + \mathrm{W} b_{ij} \leq \mathrm{W} - w_i \forall i, j \in 1, N$$
$$x_i - x_j + L.b_{ij} \leq L - l_i \forall i, j \in 1, N$$

3. Kích thước gói hàng không vượt quá thùng hàng

$$y_i \leq \mathrm{W}_k - w_i + (1 - \mathrm{f}_{ik})\mathrm{W}, \forall i \in 1, N; k \in 1, K$$

$$x_i \leq L_k - l_i + (1 - \mathrm{f}_{ik})L, \forall i \in 1, N; k \in 1, K$$

4. Mỗi gói hàng phải thuộc đúng một thùng hàng

$$\sum_{k=1}^{K} \mathrm{f}_{ik} = 1, \forall i \in 1, N$$

5. Thùng thứ k có được dùng hay không:

$$\mathrm{f}_{ik} \leq z_k, \forall i \in 1, N; k \in 1, K$$

Hàm mục tiêu:

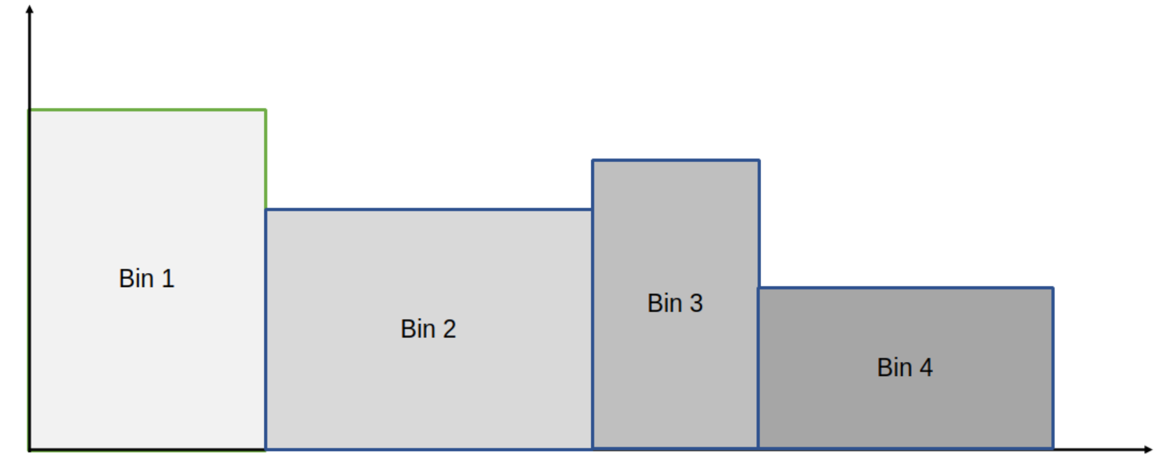
$$\sum_{k=1}^{K} c_k . z_k$$

b. Constraint programming

Ý tưởng

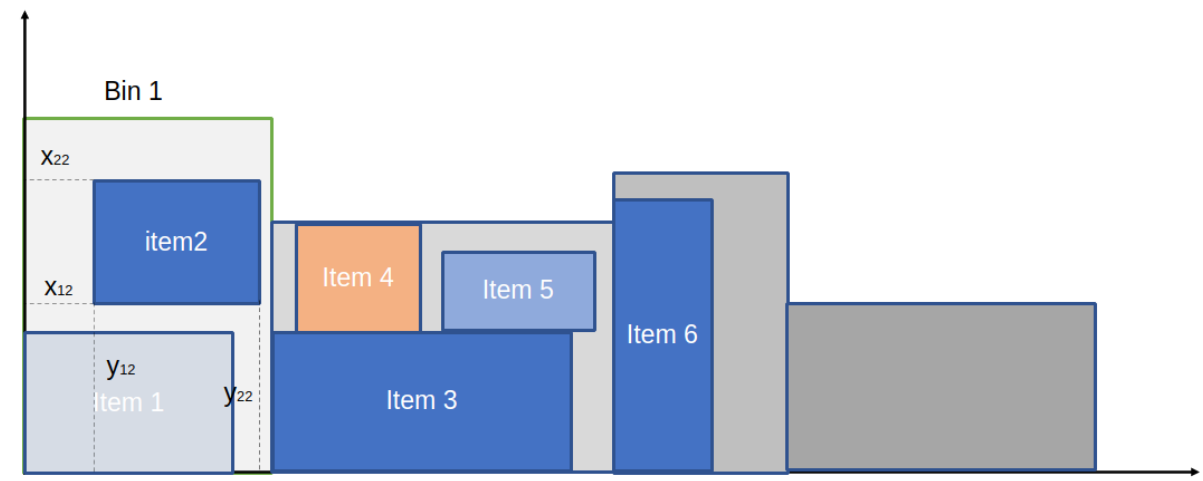
+ cho các containers xếp thành hàng trong trục toạ độ

+ sử dụng điều kiện Nonoverlaps2D của ortools giải quyết rang buộc về không gian



Các biến được sử dụng:

Biến	Ý nghĩa	Miền giá trị
$x_{1i}, y_{1i}, x_{2i}, y_{2i}$	Toạ độ của gói hàng gồm điểm cao nhất, thấp nhất trên cả 2 trục	$0 \le x_{1i} \le \sum W_j$ $0 \le y_{1i} \le \max L_j$ $\quad \forall i = 1, n; j = 1, N$
l_{ij}	Bằng 1 nếu gói hàng i nằm trong bin j	$l_{ij} = 1$ or $l_{ij} = 0$ $\forall i \in 1, N; j \in 1, K$
u_i	Bằng 1 nếu bin i được sử dụng	$u_i = 1$ or $u_i = 0$ $\forall i \in 1, N$



Các ràng buộc:

Ta có:

$$W = \max_{j \in 1,N} W_j \qquad L = \max_{j \in 1,N} L_j$$

1. Mỗi gói thuộc về một thùng hàng

$$\sum_{k=1}^{K} l_{ik} = 1, \forall i \in 1, N$$

2. Mỗi gói hàng nằm trọn trong một thùng hàng

$$\sum_{k=1}^{K} \mathrm{W}_k * l_{ik} \geq x_{1i}, \forall i = 1, N$$

$$\sum_{k=1}^{K} H_k * l_{ik} \geq y_{2i}, \forall i = 1, N$$

$$\sum_{k=1}^{K} \mathrm{W}_{k+1} * l_{ik} \geq x_{2i}, \forall i = 1, N$$

và

$$x_{1i} + \mathrm{w}_i = x_{2i}$$

$$y_{1i} + l_i = y_{2i}$$

3. Non overlap: sử dụng điều kiện NonOverlap2D của OR-tools
4. Ràng buộc giữa u và i

$$u_k = 0 \quad OnlyEnforceIf \ \sum_{i=1}^{N} l_{ik} \leq 0$$

$$u_k = 1 \quad OnlyEnforceIf \ \sum_{i=1}^{N} l_{ik} > 0$$

Hàm mục tiêu:

$$\sum_{k=1}^{K} c_k . u_k$$

c. Tabu Search

Ý tưởng:

- Mô hình hoá bởi 1 vecto n chiều xi với i= 1,…m trong đó n là các gói hàng, m là các containers

- Sử dụng chiến lược sắp xếp Shelf-algorithm, Guillotine algorithm, Maximal Rectangels Algorithm,… để sắp xếp các gói hàng vào bin

- Các chiến lược move: di chuyển một gói hàng sang bin khác, di chuyển tất cả các gói hàng sang bin khác
- Phương án được khởi tạo bằng Greedy Best fit

Mã giã của thuật toán tabu search:

Algorithm: $TabuLocalSearch$
$s_1 \leftarrow \mathrm{GENERATEINITIALSOLUTION}();$
$s^* \leftarrow s_1;$
best $\leftarrow s_1$
$\tau \leftarrow \langle s_1 \rangle;$
for $k \leftarrow 1$ **to** $MaxTrials$ **do**
    neighbors = $GETNEIGHBORS(s^*)$
    for s_k **in** $neighbors$ **do**
        if s_k $not\ in\ \tau \wedge f(s_k) < f(s^*)$ **then**
            $s^* \leftarrow s_k;$
    $\tau \leftarrow \tau :: s_{k+1};$
end
return s^*;

function $\mathrm{L-NOTTABU}(N, s, \tau)$ **return** $\{n \in N \mid n \notin \tau\}$

d. Genetic Algorithm

Ý tưởng:

- Các gen được mô hình giống trong Tabu Search và các chiến lược sắp xếp được giữ nguyên

- Cross over: sử dụng đảo đoạn bố mẹ

- Mutation: kết hợp đột biến điểm và đột biến đoạn giống như các chiến lược move của tabu search
- Phương án được khởi tạo bằng Greedy Best fit

3. Kết quả và đánh giá

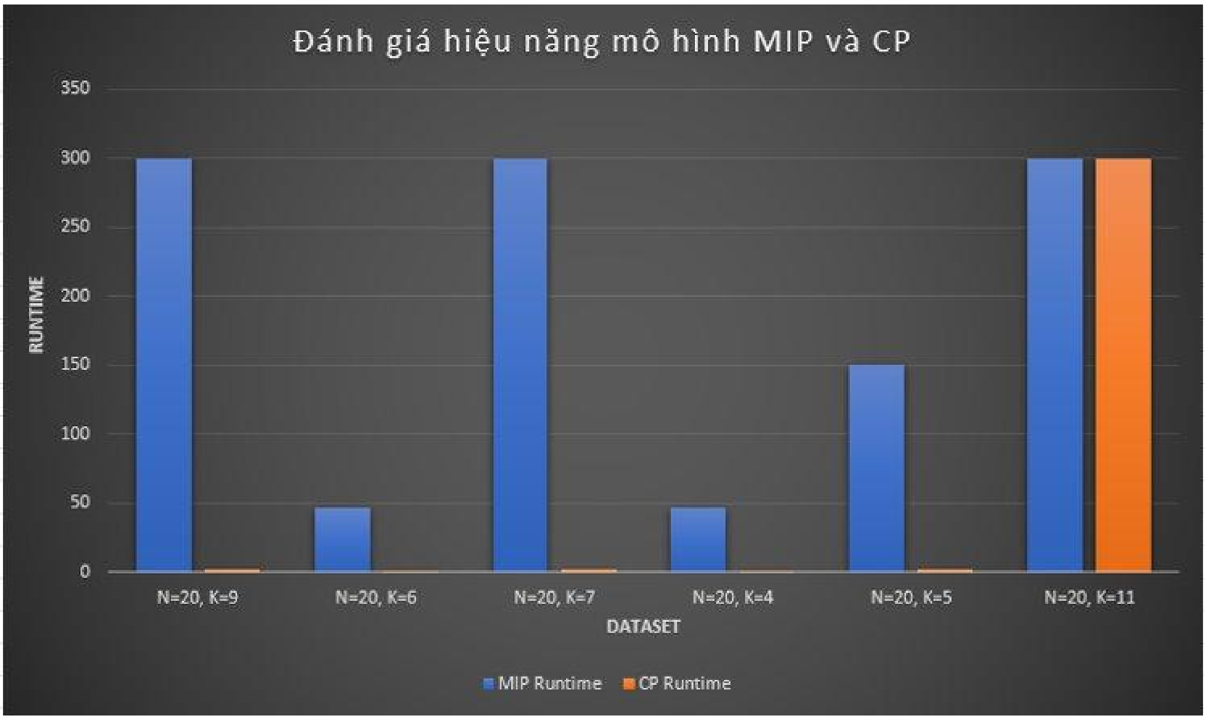
Bộ dữ liệu dùng để đánh giá bao gồm 12 trường hợp: 6 trường hợp với N = 20, trường hợp tương ứng với N = 40.

Mô hình MIP và CP trên các bộ dữ liệu:

Nếu thời gian chạy vượt quá 300s thì sẽ coi như không tìm ra lời giải.

Dataset	MIP Result	MIP Runtime	CP Result	CP Runtime
N=20, K=9	286-FEASIBLE	300.01s	286-OPTIMAL	2.41s
N=20, K=6	227-OPTIMAL	47.51s	227-OPTIMAL	1.20s
N=20, K=7	288-FEASIBLE	300.00s	288-OPTIMAL	2.61s
N=20, K=4	246-OPTIMAL	46.67s	246-OPTIMAL	0.24s
N=20, K=5	226-OPTIMAL	150.60s	226-OPTIMAL	1.69s
N=20, K=11	532-FEASIBLE	300.00s	532-FEASIBLE	300.03s

Dataset	MIP Result	MIP Runtime	CP Result	CP Runtime
N=40, K=9	315-FEASIBLE	300s	249-FEASIBLE	300s
N=40, K=6	401-FEASIBLE	300s	355-OPTIMAL	2.16s
N=40, K=7	275-FEASIBLE	300s	253-OPTIMAL	4.83s
N=40, K=4	185-FEASIBLE	300s	185-OPTIMAL	1.57s
N=40, K=5	298-FEASIBLE	300s	270-OPTIMAL	6.81s
N=40, K=11	727-FEASIBLE	300s	517-FEASIBLE	300s



Các mô hình MIP và CP đều giải ra các phương án tối ưu. Tuy nhiên, ta có thể thấy được độ vượt trội của CP so với MIP trên cả 2 tiêu chí đánh giá là thời gian chạy và số bộ dữ liệu có thể tìm được nghiệm. Cụ thể thì CP chỉ không tìm được nghiệm trên 3 bộ N = 40, k=9 và k = 11, N = 20 với k = 11. Còn MIP thì không tìm được nghuệm trên 9 bộ N = 20 với k = 7,9, 11 và toàn bộ trường hợp với N = 40. Có thể coi như với tốc độ của mình thì CP có thể giải được bài toán kích thước lớn hơn MIP trong thời gian giới hạn là 300s.

Mô hình Tabu search trên bộ dữ liệu:

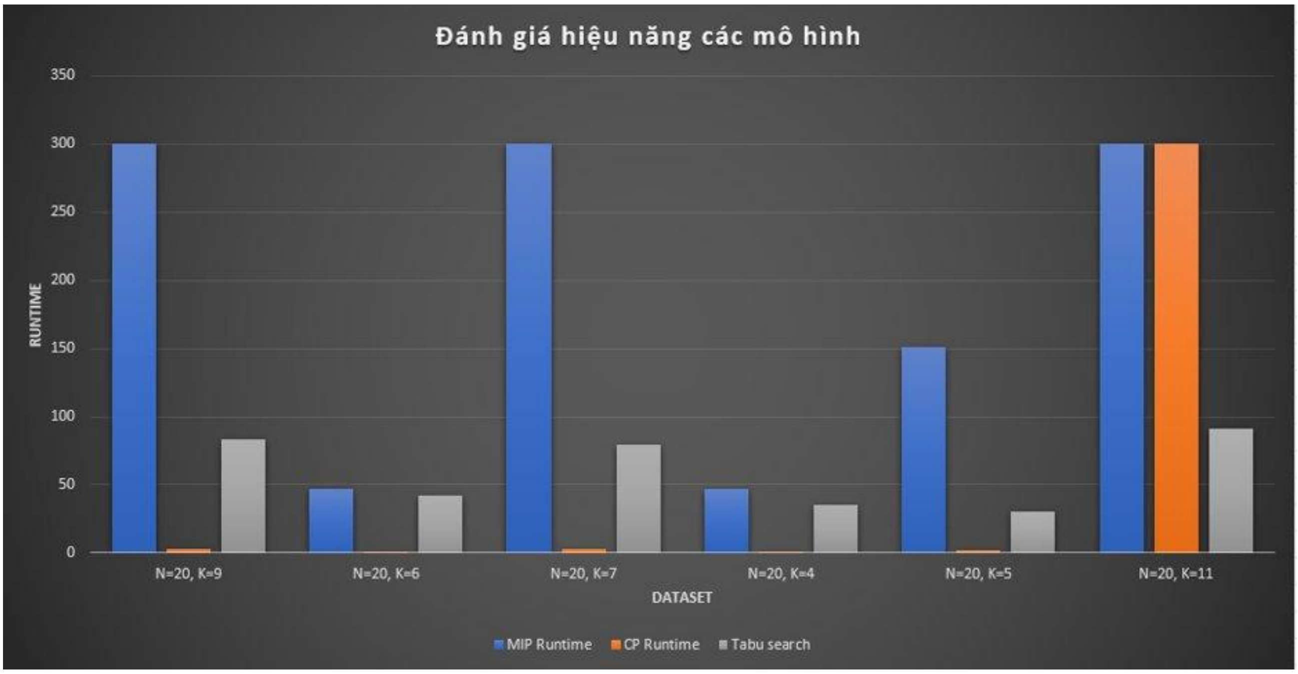
Các giá trị khởi tạo: ITERATION = 1000, TABU SIZE = 20.
Mô hình được thử nghiệm với RUN TIME = 5
Các trường hợp nền vàng là các trường hợp mà Tabu giải ra được nghiệm tối ưu

Dataset	Best Result	Worst Result	Average	Runtime
N=20, K=9	297	363	322	83s
N=20, K=6	270	281	278	42s
N=20, K=7	283	314	302	79s
N=20, K=4	246	246	246	35s
N=20, K=5	226	226	226	30s
N=20, K=11	532	590	564	91s

Dataset	Best Result	Worst Result	Average	Runtime
N=40, K=9	259	302	276	300s
N=40, K=6	355	410	383	175s
N=40, K=7	253	283	262	143s
N=40, K=4	185	185	185	201s
N=40, K=5	270	270	270	288s
N=40, K=11	530	580	556	300s

Với trường hợp N = 20, k = 4, k = 5, ta có thể thấy kết quả tồi nhất và tốt nhất bằng nhau và bằng với kết quả tối ưu của các thuật toán giải exact.
Tiếp tực với N = 40, k = 4/ 5 thì tabu search cũng cho kết quả tối ưu. Tabu search có thể cho ra các phương án chấp nhận được trên toàn bộ tập thử nghiệm, đó là điều mà cả MIP và CP đều không làm được. Đồng thời Tabu cũng cho kết quả tối ưu với các trường hợp có độ phức tạp tính toán thấp.

Sau đây là so sánh thời gian chạy và hàm mục tiêu của mô hình Tabu search và 2 thuật giải exact:



Mô hình GA trên bộ dữ liệu:

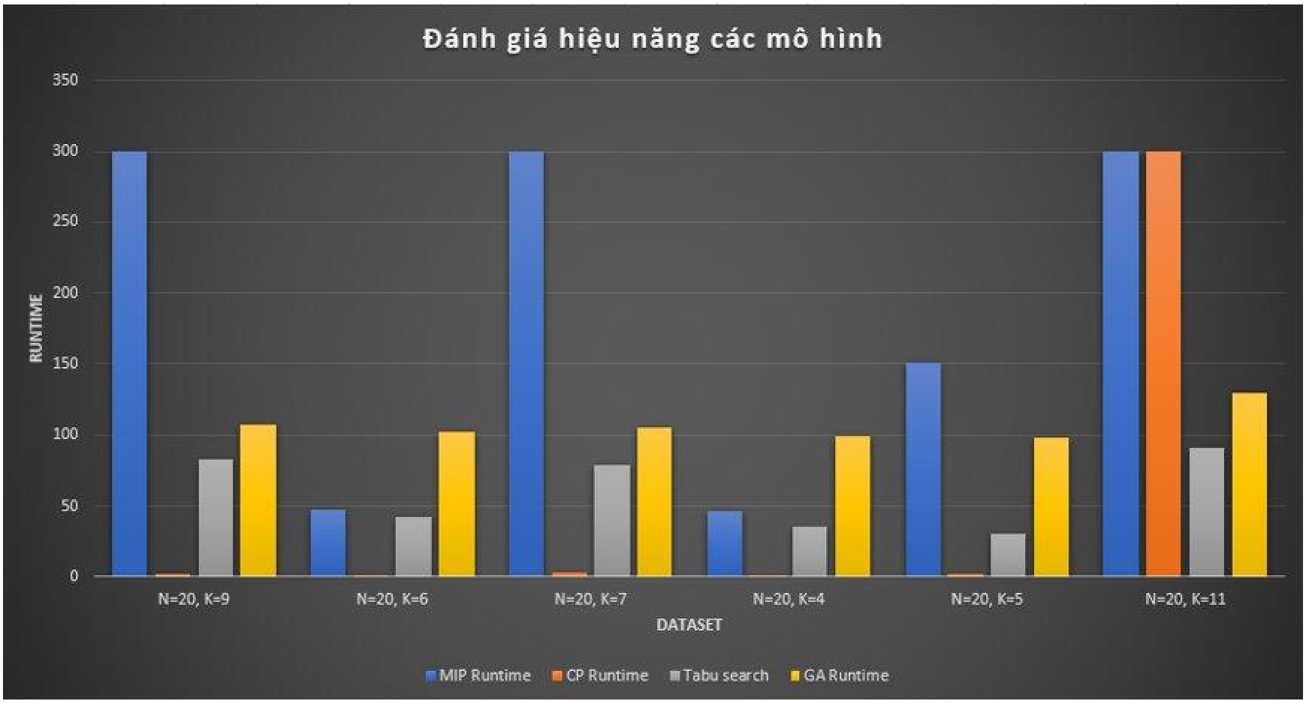
Population = 100, mutation = 0.1, cross_over = 0.8

Dataset	Best Result	Worst Result	Average	Runtime
N=20, K=9	335	363	349	107s
N=20, K=6	289	304	296	102s
N=20, K=7	307	321	313	105s
N=20, K=4	246	261	250	99s
N=20, K=5	226	226	226	98s

N=20, K=11	639	673	658	130s

Kết quả của GA có đôi chút kém hơn so với tabu search ở cả giá trị hàm mục tiêu và thời gian tính toán. GA chỉ cho ra một kết quả tối ưu với N = 20, k = 5

So sánh cả 4 mô hình trên bộ dữ liệu N = 20



4. Kết luận

- MIP và CP giải bài toán cho ra kết quả tối ưu.
- Đối với một số bộ dữ liệu lớn, MIP và CP không cho kết quả.
- Tabu search và GA giải ra các phương án chấp nhận được trong khoảng thời gian khá ngắn.
- Các thuật giải heurestic cho ra kết quả đối với toàn bộ tập thử nghiệm và ra nghiệm tối ưu với các bộ dữ liệu nhỏ.

So sánh các mô hình:

Các thuật giải chính xác: MIP < CP

Các giải thuật heurestic: GA < Tabu

Đặc biệt với các bộ dữ liệu nhỏ (4 mô hình đều cho ra kết quả tối ưu): MIP < GA < Tabu <CP

Danh mục tài liệu tham khảo

1] D. Pisinger, M.M. Sigurd, Using decomposition techniques and constraint programming for solving the two-dimensional bin packing
problem, Technical Report DIKU-03/01, Department of Computer Science, University of Copenhagen, Denmark, 2002.

[2] Pisinger, D., & Sigurd, M. (2005). The two-dimensional bin packing problem with variable bin sizes and costs. Discrete Optimization, 2(2), 154–167.

[3] Liu, Y., Chu, C., & Wang, K. (2011). A dynamic programming-based heuristic for the variable sized two-dimensional bin packing problem. International Journal of Production Research, 49(13), 3815–3831.

[4] M. Berger, M. Schröder, K.-H. Küfer A constraint programming approach for the two-dimensional rectangular packing problem with orthogonal orientations

[5] A. Lodi, S. Martello, D. Vigo, Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems, INFORMS J.Comput. 11 (1999) 345–357.

[6] Solving a MIP Problem | OR-Tools - Google Developers.

[7] J.E. Beasley, OR-library, 2004, http://mscmga.ms.ic.ac.uk/info.html.