Kasetsart J. (Nat. Sci.) 37 : 514 - 522 (2003)

A Hybrid Genetic Algorithms and Tabu Search for Solving an Irregular Shape Strip Packing Problem

Kittipong Ekkachai[1] and Pradondet Nilagupta[2]

ABSTRACT

This paper presents a packing algorithm to solve an irregular shape strip packing problem. The polygons are packed in strip material, which limits a bin width but not limits a bin length. The objective of a packing algorithm is to minimize a bin length. This paper proposes a hybrid method of genetic algorithms and tabu search to solve a packing problem. Initially, a hybrid method behaves like regular genetic algorithms. After that, a hybrid method is self adapting until it behaves like a tabu search. This paper compares results generated by a hybrid method with genetic algorithms and tabu search. The experiment data is the 13 pieces convex-type polygon and duplicate, triplicate and quadruplicate. The experiment shows that a hybrid method generates better result than genetic algorithms about 2.56% to 4.69% and a hybrid method generates better result than tabu search about 0.26% to 2.78%.

Key words: nesting problem, no-fit polygon, genetic algorithms, tabu search, meta-heuristics

INTRODUCTION

Packing problems are optimization problems concerned with finding a good arrangement of multiple items in large containment. This paper presents a packing algorithm for irregular shape polygons. Polygons are packed in strip material. Strip material has an important characteristic, which limit a bin width but not limit a bin length. The objective of a packing algorithm is to find a layout which minimized a bin length. This problem can be found in textile industry and metal cutting. The packing problem is one of a non-deterministic polynomial time (NP) problem. The NP problem requires a number of computational steps that grows exponentially with the problem size to find optimal solution. Currently, it dose not have an efficient algorithm to find optimal solution. Alternative approaches find near optimal solution. Many researchers use heuristics and meta-heuristics method to solve irregular shape strip packing problem.

Hybrid genetic algorithms (GA) and bottom-left (BL) heuristic was developed (Jakob, 1994). Polygons are bracketed by rectangular shape. BL is used to pack each polygon to a bottom-left most position of a container. GA is used for finding order of polygons. This method is simple and fast but it is not efficient for a complex polygon. Hybrid binary tree representation and GA was developed (Bounsaythip and Maouche, 1997). A Binary tree represents a layout and leaves represent polygons. Two leaves can be combined to a larger polygon. This method use genetic algorithms to move leafs to find a good solution. Grid representation and GA was developed (Rattanapan

[1] Department of Electronic, National Electronics and Computer Technology Center, Pathumthani 12120, Thailand.
[2] Department of Computer Engineering, Faculty of Engineering, Kasetsart University, Bangkok 10900, Thailand.

Received date : 07/07/03 Accepted date : 26/11/03

and Dagli, 1997). Polygons are represented by grid cells. This method uses transformation matrix such as translation and rotation to mutation and crossover process. GA move all grid cells until it found a good solution. The advantage of this method is all feasible solution can be found because this method dose not reduces the input dimensionality. However, the quality of result proportion to a resolution of a grid cell. And a higher resolution requires a lot of computational times and memory. TOPOS algorithm (Oliveira *et al.*, 2000) was developed. TOPOS algorithm uses a greedy method and several criteria to find a best placement. TOPOS tries 126 runs in different criteria and selects a best solution. Meta-heuristics and no-fit polygon was developed (Kendall, 2000). Polygon order represents a layout. No-fit polygon determine feasible position that two polygon can be nested without overlapping area. This method selects a position that generates minimum convex hull and combine two polygons to be a new polygon. A new polygon is a stationary polygon and combines with a next polygon. The process is running until all of polygons are nested. This method use a various type of meta-heuristics such as GA, simulated annealing, tabu search (TS), ant algorithm, and memetic algorithm to find a best polygon order.

This paper presents a hybrid method that combine GA and Tabu Search (TS) apply with no-fit polygon to solve an irregular shape strip packing problem. This method combines advantages of two meta-heuristics. Initially, the algorithm behaves like a regular GA. Once, a solution saturate, a hybrid method will reduce population and increase mutation probability. A hybrid method is self adapting until it behaves like TS.

MATERIALS AND METHODS

Packing algorithm

The outline of packing algorithm has the following structure.

1. A solution of a problem is represented by a permutation of a polygon order which is initialized by a random method.
2. Calculate a no-fit polygon from a first polygon and a second polygon.
3. Select a position on a boundary of a no-fit polygon that generates minimum cost. The cost that generated by position is depend on placement criteria. The selected position is a position of a second polygon that touches a first polygon.
4. Combine two polygons to be a new polygon.
5. Set a new polygon to a stationary polygon and calculate a no-fit polygon with a next polygon. Repeat step 2 to step 4 until all polygons are nested.
6. Calculate a layout length.
7. Apply meta-heuristics to find a best polygon order.

No-fit polygon

No-fit polygon is an efficient tool to determine feasible positions that two polygons are touched without overlap. Figure 1 shows a no-fit polygon of polygon A and polygon B. Polygon A is a stationary polygon and polygon B is an orbit polygon. Dash line is a no-fit polygon. If a reference point of polygon B is placed in an interior of a no-fit polygon, then polygon B intersects polygon A. If a reference point of polygon B is placed on a

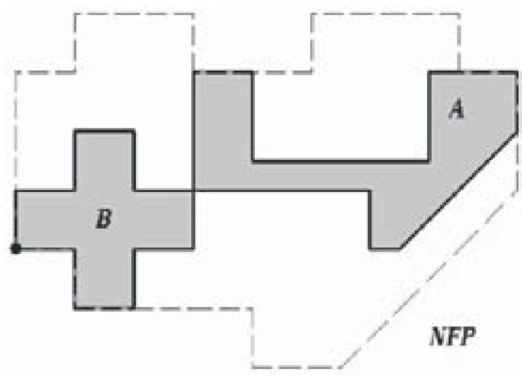

Figure 1 No-fit polygon.

boundary of a no-fit polygon, then polygon B touches polygon A. The boundary of a no-fit polygon is feasible positions of polygon B that can be nested with polygon A.

A computational method to calculate a no-fit polygon from convex polygons is called a Minkowski's difference. If A and B are two sets of edges in two polygons, then a no-fit polygon can be calculate from equation

$$NFP_{AB} = A \oplus -B$$
$$A \oplus B = \{a+b \mid a \in A, b \in B\}$$
$$-A = \{-a \mid a \in A\}$$

a, b is the edge in A, B

Figure 2 shows an example of a method to calculate a no-fit polygon. Polygon A has a counter clockwise orientation. Polygon B has a clockwise orientation. No-fit polygon is a set of edges that are combined from two polygons.

A method to calculate the no-fit polygon from non-convex polygons must include some additional processes shown in a Benell's paper (Benell *et al*., 2001) and a Computational Geometry in C (O'Rourke, 1998)

Placement criteria

In recent work (Kendall, 2000), a convex hull area of two polygons was used for determining a position of an orbit polygon that used to nest with a stationary polygon. A position that generated minimum convex hull area was selected. In our work we tried to use several criteria. We found that a calculation of a convex hull is required more computational time than other criteria. This paper uses a multi-phase criterion to select a best placement of an orbit polygon. The first phase of a packing algorithm uses a MinCenterDistance criterion that shows in figure 3(a). MinCenterDistance is a placement criterion, given a minimum center distance of two rectangular enclosure polygons. The second phase of a packing algorithm uses a MinLength criterion shown in figure 3(b). MinLength is a placement criterion, given a minimum length of two polygon layout. The first halves of all polygons use a MinCenterDistance criterion and the second halves of all polygons use a MinLength criterion. In our research the multi-phase criterion generated a better result than other criteria. However, in some situations a placement criterion can not select a best position, because that position generates a placement outside a container. Packing algorithm is check placement width. If a placement width is longer than a bin width, then this placement is invalid and an algorithm will select a next valid position.

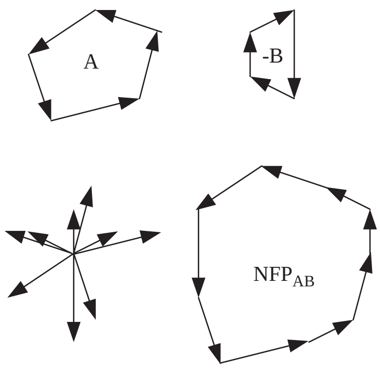

Figure 2 A method to calculate a no-fit polygon.

Evaluation

Evaluation value or cost is a layout length. If a placement lie on a horizontal, then an evaluation value is a distance between right most and left

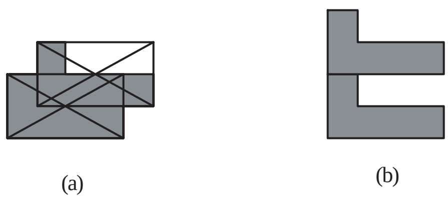

Figure 3 The two placement criteria are MinCenterDistance (a) and MinLength (b).

most point of a layout. If a placement lie on a vertical, then an evaluation value is a distance between top most and bottom most point of a layout.

Hybrid method of GA and TS

A hybrid method of meta-heuristics comes from an inspiration to combine advantages of two meta-heuristics GA and TS. The advantage of GA is that a solution improves rapidly during initial phase. But in a later phase, a solution improves slowly. TS is a search method that hold a single solution each time. TS find a new solution from neighborhood of a current solution. A solution improves rapidly but a quality of a solution depends on a random starting point. Main search in a hybrid method is GA and mutation module is TS. Initially, a hybrid method behave like GA with low mutation probability. Once, a solution saturate, a hybrid method will reduce population and increase mutation probability. A hybrid method is self adapting until population reduce to 1 and mutation probability increase to 1.0, then a hybrid method behave like TS.

In Figure 4 and Figure 5 show structure of a hybrid method. Main search is GA that replaces mutation module with TS module. We call a polygon order as an individual. Population contains a set of individuals. Initially, a population size *pop_count* is equal to a parameter *Pop* and a mutation probability *Pm* is equal to a parameter *Pm0*. A parameter *Pop* is an initial value of population size and a parameter *Pm0* is an initial value of mutation probability. An initial population is created with random method and each individual has its own a tabu list *TL*. An initial *TL* is empty. In a crossover section, a hybrid method selects pairs of individuals *Vc1* and *Vc2* with random method. The number of pairs that use in a crossover process depends on a probability *Pc*. When a crossover process finished, it return two offspring *O1* and *O2*. A hybrid method replaces two worst individual by these offspring. In a mutation section,

```
procedure Hybrid GA
begin
    t = 0
    Pm ← Pm0
    pop_count ← Pop
    Initialize (Population, pop_count)
    Evaluate Population
    while (not termination - condition) do
    begin
        if  (no improvement for R generation)
        begin
            Reduce_Pop(pop_count ,Dec)
            Pop ← the best pop_count from
              Population
            Increase_Mutation(Pm,Inc)
        end
        select Vc1 ,Vc2 from Population with Pc
        Crossover(Vc1,Vc2,O1,O2)
        Population ← the best of (Population U
          O1,O2 )
        select Vc from Population with Pm
        TSMutate(Vc,TL)
        t = t+1
        end
end
```

Figure 4 Structure of hybrid GA.

```
procedure TSMutate(Vc,TL)
begin
    select a new point Vn that is
            the best of  the  neighborhood of Vc
    if ((Vn not in TL)or
            (Vn is better than the best solution))
    then Vc ← Vn
    UpdateTabuList(TL)
end
```

Figure 5 Structure of TS mutation module.

a hybrid method selects an individual *Vc* by random. The number of individuals that use in mutation process depends on a probability *Pm*. Each individual has its own *TL* that use in a *TSMutate* module. In *TSMutate*, a TS processes runs only one time. A *TSMutate* module starting by selects a best individual *Vn* from candidate neighbors of *Vc*. If *Vn* is found in *TL* and *Vn* is not better than current best solution, then a TS process selects a second place individual that invalid in above condition. Otherwise a TS process selects *Vn* to be a current solution *Vc*. Number of candidate neighbors is equal a neighborhood size *NH*. A *TSMutate* Module updates *TL* and return a new *Vc*. A hybrid method replaces an old *Vc* with a new *Vc* that return from a *TSMutate* module. If a solution dose not improve in R generation, then a hybrid method reduce *pop_count* with multiply by decreasing rate *De*c and increase *Pm* with multiply by increasing rate *Inc*. When *pop_count* reduces to 1 and Pm grows to 1.0, a hybrid method behaves like TS. The characteristics of the hybrid method that use in this work are described below.

- A crossover type is Order Crossover (OX)
- Neighbors of a current solution are generated by swapping two polygons in a current solution.
- Tabu lists keep pairs of polygons that recently swapped. The size of tabu list is equal to a parameter *T_Size*

All parameters that use in the hybrid method are describe below

Pop is an initial population size.
Pc is a crossover probability.
Pm0 is an initial mutation probability.
R is a limited number of generations that solution not improves.
Dec is a decreasing rate of a population size
Inc is an increasing rate of a mutation probability
NH is a neighborhood size
T_Size is a tabu list size

RESULTS AND DISCUSSION

The hybrid method is compare with GA and TS. The characteristics of GA that use in an experiment are the following

- A crossover type is OX.
- A mutation type is inversion. An inversion process is select a part of individual with random method and reverse order in that part
- A selection process use roulette wheel to select a new population from an old population and new individuals that generate with a crossover and a mutation process

Parameters that use in GA are the following.

Pop is a population size.
Pc is a crossover probability.
Pm is a mutation probability

The characteristics of TS that use in an experiment are the following

- Neighbors of a current solution are generated by swapping two polygons in a current solution.
- Tabu lists keep pairs of polygons that recently swapped.

Parameters that use in TS are the following.

NH is a neighborhood size
T_Size is a tabu list size

The experiment was executed 10 times per data set and each set of parameters. We select a parameter that shows a best result in each method to be compared. The computational tests were run on PC with Intel Celeron 2.0 GHz Ram 256 Mb and run on Windows 2000. Runtime in each data set is estimate for 2000 evaluations.

Data sets

This paper use a data set that appears in Kendall research (Kendall, 2000). The data set have 13 different pieces convex polygon and packing to strip material. In this paper we test a

Kendall data set and duplicate, triplicate and quadruplicate. The numbers of polygons are 13, 26, 39 and 52 polygons. Table 1 shows the detail of data sets.

Table 2 shows the experiment result of three meta-heuristics including GA, TS, and a hybrid method (GA/TS). TS generates better result than GA in all data sets. And a hybrid method generates better result than both GA and TS shown in table 3. Table 4 shows a best parameter in each method and each data set that found the experiment. Figure 6 shows material utilization that is generated by three meta-heuristics. Figure 7 shows the improvement of layout length in Kendall52 data set.

CONCLUSION

The experiment shows that a hybrid method generates better result than both GA and TS. The experiment also shows TS generates better result than GA. In figure 7 shows TS generates better result than other method in initial phase, but it generates worse in later phase. A hybrid method behaves like genetic algorithms and result are worse than tabu search in initial phase. But in later phase, a hybrid method self adapting control parameter that make result improve rapidly. In last phase, a hybrid method behaves like TS and generates result better than regular TS. Because a hybrid method use spread searching at initial phase but tabu search use random start at initial phase. The methods that we choose for packing problems

Table 1 Detail of data sets.

Data set	Polygon type	Number of polygon	Runtime (Sec)	Bin width
Kendall13	13	13	40	6240
Kendall26	13	26	180	6240
Kendall39	13	39	420	6240
Kendall52	13	52	1000	6240

Table 2 Average layout length and average material utilization that found in the experiment.

	Method	Average layout length	Average material utilization (%)
Kendall13	TS	23890.04	79.3337
	GA	24416.61	77.6254
	GA/TS	23797.56	79.6128
Kendall26	TS	48728.39	77.7836
	GA	49872.30	75.9766
	GA/TS	48591.19	77.9892
Kendall39	TS	74352.03	76.4390
	GA	75962.44	74.8151
	GA/TS	73906.69	76.8913
Kendall52	TS	99796.87	75.9244
	GA	102848.30	73.6761
	GA/TS	98248.98	77.1317

are trade-off between material cost and computational time. If we have a few times, then we will choose tabu search. But if we want a better layout, then we will choose a hybrid method. Figure 8 shows the best layout of Kendall13 that found in the experiment. A layout length is 22446 and a material utilization is 84.385%. Figure 9 shows the best layout of Kendall52 that found in the experiment. A layout length is 94647.52 and a material utilization is 80.07%. This packing algorithm was done for non-convex polygon also. The non-convex data set that appears in other research (Oliveira, 2000) was tested. Packing algorithms without any rotation use tabu search

Table 3 A hybrid method generates better result than GA and TS shown in percentage.

	Better result that generated by a hybrid method	
	GA	TS
Kendall13	2.56 %	0.35%
Kendall26	2.65%	0.26%
Kendall39	2.78%	0.59%
Kendall52	4.69%	2.78%

Table 4 The parameters are used in the experiment.

	Method	Parameter
Kendall13	TS	NH = 50, T_Size = 50
	GA	Pop=40,Pc=0.6,Pm=0.1
	GA/TS	Pop = 50,Pc=0.7,Pm0=0.01, R=4,Dec=0.7,Inc=1.4, NH=20,T_Size=50
Kendall26	TS	NH = 30, T_Size = 100
	GA	Pop=30,Pc=0.7,Pm=0.1
	GA/TS	Pop = 50,Pc=0.7,Pm0=0.05, R=4,Dec=0.7,Inc=1.4, NH=5, T_Size =30
Kendall39	TS	NH = 20, T_Size = 50
	GA	Pop=20,Pc=0.8,Pm=0.1
	GA/TS	Pop = 50,Pc=0.7,Pm0=0.01, R=4,Dec=0.7,Inc=1.4, NH=20, T_Size =50
Kendall52	TS	NH = 5, T_Size = 50
	GA	Pop=30,Pc=0.7,Pm =0.1
	GA/TS	Pop = 50,Pc=0.7,Pm0=0.05, R=4,Dec=0.7,Inc=1.4, NH=5, T_Size =30

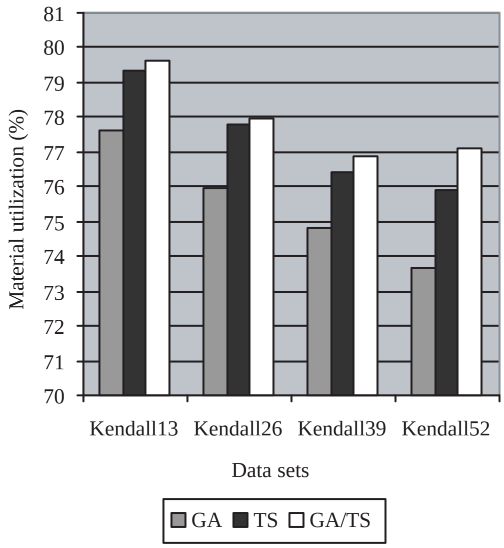

Figure 6 Material utilizations that are generated by three meta-heuristics.

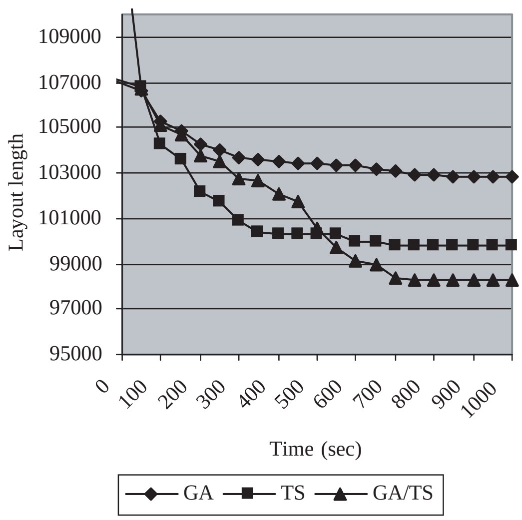

Figure 7 An improvement of layout length in Kendall52 data set.

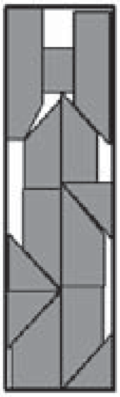

Figure 8 The best layout found in Kendall13.

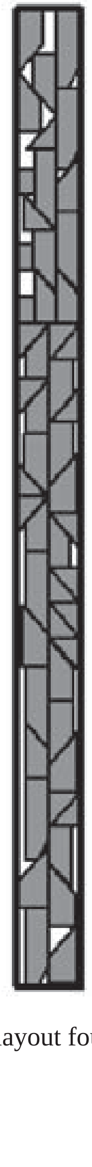

Figure 9 The best layout found in Kendall52.

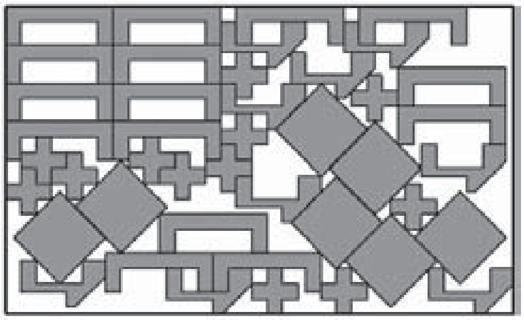

Figure 10 The best layout found in non-convex data set.

and run about 2000 evaluations. The best layout is showed in Figure 10. A layout length is 66 and a material utilization is 60.455%.

For future work a parallel algorithms of meta-heuristics may be introduced in packing problem to reduce run time. Some pre-process can reduce unnecessary computation such as Kendall13 layout. If we duplicate Kendall13 layout 4 times, then a result is better than one Kendall52 layout.

LITERATURE CITED

Benell J.A., Dowsland K.A. and W.B. Dowsland. 2001. The Irregular Stock Cutting Problem – a new procedure for deriving the no-fit polygon. **Computer and Operations Research** 28 : 271-287

Bounsaythip, C. and S. Maouche. 1997. Irregular shape nesting and placing with evolutionary approach. Proceeding of the **IEEE International Conference on Systems, Man and Cybernatics** 4 : 3425-3430

Jakobs S. 1994. On genetic algorithms for the packing of polygons.**European Journal of Operational Research** 88 : 165-181

Kendall, G. 2000. **Applying Meta-heuristic Algorithms to the Nesting Problem Utilizing the No Fit Polygon**. Ph.D. Thesis. University of Nottingham.

Oliveira, J.F., A.M. Gomes and S. Ferreira. 2000

TOPOS A new constructive algorithms for nesting problems. **OR Spektrum** 22 : 263-284

O'Rourke, Joseph. 1998. **Computational Geometry in C**. Cambridge University Press.

Ratanapan, K. and C.H. Dagli. 1997. An object – based evolutionary algorithm for solving rectangular piece nesting problem, pp. 989-994. *In* **Proceeding of the IEEE Conference on Evolutionary Computation**. Piscataway, NJ, USA