



Proceedings of GLOGIFT 12
July 30 – August 1, 2012
University of Vienna, Austria
pp. 863-872

A Particle Swarm Optimization for Polarizer Cutting Plan Scheduling Problem

Shih-Tang Lo¹, Der-Fang Shiau² and Tung-Ju Wu³

Abstract

Polarizer cutting plan (PCP) is based on the month of customer orders, demand for a variety of different specifications of the polarizer, and selects the appropriate cutting die, cutting sequence and arrangement of the polarizer cutting plan. This study proposes a particle swarm optimization algorithm for minimizing the polarizer material usage to save the consumption cost of the materials. Furthermore, according to the rule of artificial way, we utilize the computer system automation to optimize the arrangement of cutting plan, material savings and the inventory reduction without increasing any fixed and variable costs. Experimental results demonstrate the robustness of the proposed PSO algorithm in terms of solution quality.

Keywords: Cutting Mold, Polarizer cutting plan, Particle Swarm Optimization, Scheduling

Introduction

Polarizer cutting plan (PCP), is based on the month of customer orders, demand for a variety of different specifications of the polarizer, and selects the appropriate cutting die and cutting sequence for the arrangement of the polarizer cutting plan [1-3]. A better cutting plan not only can effectively save the consumption of materials, but also shorten the production time and cost. PCP problem was never discussed in the previous literature. However, the current production polarizer industries are facing important issues [4]. This study proposes a particle swarm optimization algorithm for minimizing the polarizer material usage to save the consumption cost of the materials.

Problem Description

The Problem of Polarizer Cutting

Polarizer cutting plan scheduling problem (PCPSP) is different from the general products when cutting any angle arrangement (paper, wood, glass ... etc.) to obtain the optimal material utilization. In reality, there are some products in addition to outside dimensions, but also must consider the limitations of the product characteristics, such as the polarizer, bright enhancement film, dual the brightness enhancement film. They need to consider the internal angle when the cutting

-
1. Department of Management of Information, Kun Shan Univ. Taiwan
Email: edwardlo@mail.ksu.edu.tw
 2. Department of Management of Information, Fooyin University. Taiwan
Email: ft041@mail.fy.edu.tw*Corresponding author
 3. Department of Engineering Science National Cheng Kung University, Taiwan
Email: N97991105@mail.ncku.edu.tw

is processed. The angle is necessary and important to obtain the characteristics of the product (such as the corrected color). Thus, such products call the product limitation for the angle, shown in Figure 1.

The Knife Mold of Polarizer Cutting

The main production tool is cutting knife mold in the cutting process. Since polarizer belongs to angle-limited products; therefore, when customers have new demands for different size productions and a new knife mold for the size product is built for this customer. Figure 2 shows the single knife mold of polarizer cutting and figures 3 and 4 show the 2 in 1 and 3 in 1 of composite knife mold, respectively. In the time of cutting, the composite knife mold can cut a variety of different sizes polarizer to maximize the volume of material.

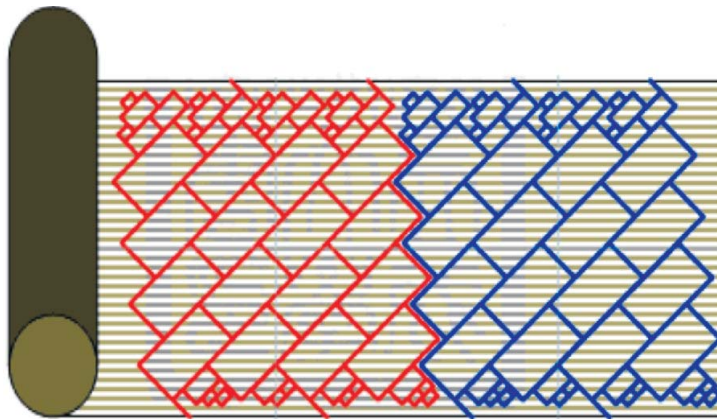


Figure 1: The Diagram of Cutting Angle of Polarizer Knife Mold

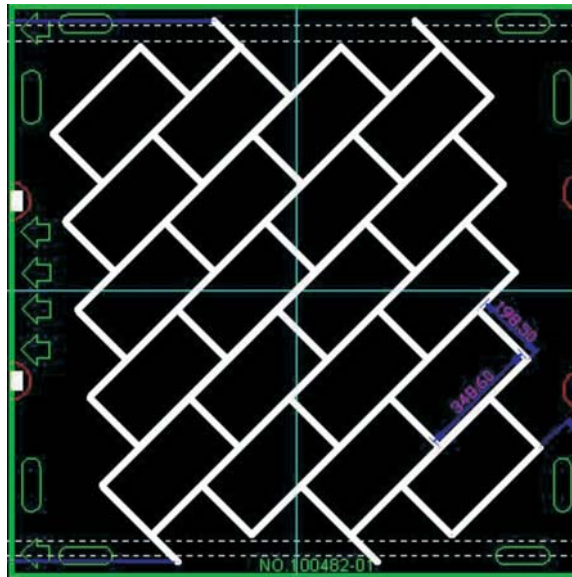


Figure 2: Knife Mold of Polarizer Cutting (single)

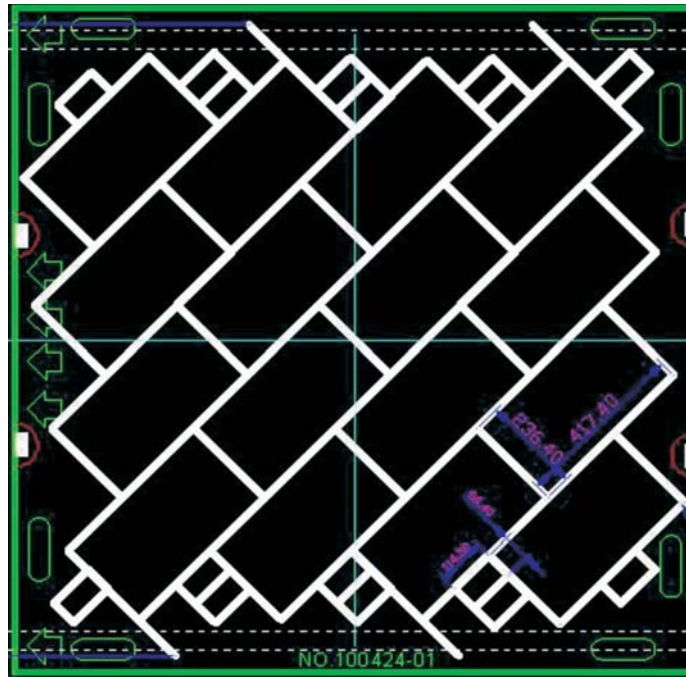


Figure 3: Knife Mold of Polarizer Cutting (2 in 1)

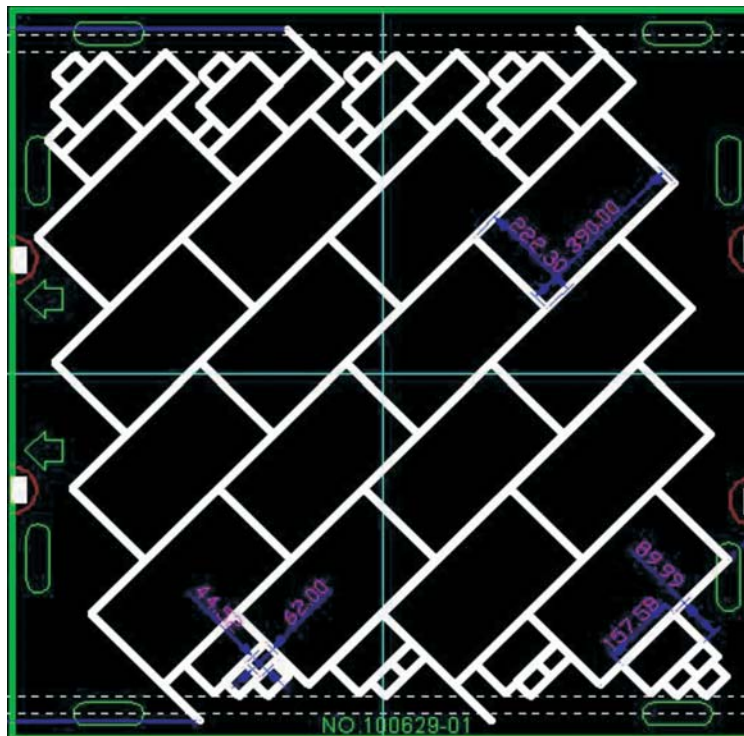


Figure 4: Knife Mold of Polarizer Cutting (3 in 1)



Polarizer Cutting Plan

Polarizer cutting plan (PCP), is based on the demand of the month a variety of finished products; i.e., this demand is based on total demand for the current month minus the existing inventory. Therefore, how to optimally arrange the order of knife molds and determine the quantity of each cutting that minimize the usage of the polarized materials, and making material costs can be more economical. There are n knife molds, $N = \{1, 2, \dots, n\}$, to be used during the cutting process. Table 1 shows the product demand for the current month (simulated data). The objective is to find an order of knife molds that minimizes the usage of materials to satisfy the demands of customers. A PCP process is described as follows:

Step 1: According to the knife mold selected in N , the following equation describes the required throughput, and table 2 shows the result of an example.

$$\text{Required throughput} = \text{Demand} / \text{Yield rate} \quad (1)$$

Step 2: Equation 2 calculates the number of pieces per meter according to the selected knife mold

Number of pieces per meter = No. of pieces / (valid length + GAP) * 1000mm, GAP is the length between two cuts (2)

$$27'' = 2 \text{ pieces} / (919.83\text{mm} + 7\text{mm}) * 1000\text{mm} = 2.1579 \text{ pieces}$$

$$23.6'' = 4 \text{ pieces} / (919.83\text{mm} + 7\text{mm}) * 1000\text{mm} = 4.3158 \text{ pieces}$$

$$7'' = 6 \text{ pieces} / (919.83\text{mm} + 7\text{mm}) * 1000\text{mm} = 6.4737 \text{ pieces}$$

$$5'' = 6 \text{ pieces} / (919.83\text{mm} + 7\text{mm}) * 1000\text{mm} = 6.4737 \text{ pieces}$$

Step 3: Therefore, we can calculate the requirement of materials for different sizes of products based on Eqs. 1 and 2. Equation 3 calculates the requirement of materials (meter) and table 3 shows the results.

$$\text{Requirement of materials (meter)} = \text{Throughput} * \text{number of pieces} \quad (3)$$

Step 4: According to the step 3, equations 4 and 5 calculates the cutting quantity and remaining quantity, respectively. Table 4 shows the results.

$$\text{Cutting quantity} = \text{Requirement of materials (meter)} * \text{number of pieces} * \text{Yield rate} \quad (4)$$

$$\text{Remaining quantity} = \text{Demand} - \text{Cutting quantity} \quad (5)$$

Step 5: Stop the process if satisfy the product demand; go to step 1, otherwise.

Table 1: Product Demand Form

Material no.	Product name	size	Yield rate	Demand
FCMF09305301	M270H1-L01-F	27.0	0.935	31947
FCMF05605301	M236H1-L07-F	23.6	0.955	228599
FCPF08205301	215FA01-F	21.5	0.975	94048
FILF05305301	M185GW01-F	18.5	0.965	93084
FCPF15705301	POL-C-090JD01-F	9.0	0.965	75956
FILF02705301	8D2- F	8.0	0.965	31028
FILF00705301	7D-F	7.0	0.970	125174
FHSF01005301	I0706-F	7.0	0.970	99429
FILF04805302	5DW- F	5.0	0.750	26484
FHSF10205301	I0503-F	5.0	0.750	135911
FCMR09305401	M270H1-L01-R	27.0	0.935	25929
FCMR05605401	M236H1-L07-R	23.6	0.955	204085
FCMR08205401	M215H1-L01	21.5	0.975	124901
FILR05305401	M185GW01-R	18.5	0.965	51915
FCHR02705401	8D- R	8.0	0.965	41875
FILR00705401	7D- R	7.0	0.970	105984

Table 2: The Example of Required Throughput

knife mold.	Material no	size	Yield rate	Demand	Throughput
5IRN75002F	FCMF09305301	27.0	0.935	31947	34168
	FCMF05605301	23.6	0.955	228599	239371
	FHSF01005301	7	0.975	99429	102505
	FHSF10205301	5	0.965	135911	181215

Table 3: Requirement of Materials

knife mold.	Material no	size	Yield rate	Demand	Throughput	Number of pieces	Requirement of materials (meter)
5IRN75002F	FCMF09305301	27.0	0.935	31947	34168	2.1579	15834
	FCMF05605301	23.6	0.955	228599	239371	4.3158	15834
	FHSF01005301	7	0.975	99429	102505	6.4737	15834
	FHSF10205301	5	0.965	135911	181215	6.4737	15834

Table 4: Cutting Quantity and Remaining Quantity

knife mold.	Material no	size	Yield rate	Demand	Number of pieces	Requirement of materials (meter)	Cutting quantity	Remaining quantity
5IRN75002F	FCMF09305301	27.0	0.935	31947	2.1579	15834	31947	0
	FCMF05605301	23.6	0.955	228599	4.3158	15834	65261	163338
	FHSF01005301	7	0.975	99429	6.4737	15834	99429	0
	FHSF10205301	5	0.965	135911	6.4737	15834	76878	59033

Particle Swarm Optimization

The particles swarm optimization (PSO) is first proposed by Kennedy and Eberhart [5]. In PSO, a swarm of particles spread in the space and the position of a particle presents a solution. Each particle would move to a new position decided with the global experience and the individual experience heading for the global optimum. Luo *et al.* [6] used PSO to solve the resource-constrained project scheduling problem; they showed that PSO is applicable to various combinatorial and scheduling problems. Furthermore, Salman *et al.* [7] presented PSO for task assignment problem; they also demonstrated that the proposed PSO-based algorithm solution quality is better than that of genetic algorithm (GA) in most of cases. Moreover, the PSO algorithm runs faster as compared with GA. Tseng and Liao [8] demonstrated that the proposed PSO algorithm outperforms over GA and ant colony system (ACO) for the flow shop scheduling problem. More recently, other applications of PSO for scheduling problem can be found in [9, 10]. To the best of our knowledge, this study is the first attempt in proposing a PSO for Polarizer cutting plan problem.

The PSO algorithm is a multi-agent general meta-heuristic method, and can be applied extensively in solving many difficult problems. The PSO consists of a swarm of particles in the space; the position of a particle is indicated by a vector which presents a solution (a schedule). PSO is initialized with a population of Np random particles and searches for the best position (solution or schedule). In every generation or iteration, the local best and global best are determined through evaluating the performances in terms of the fitness values of current population of particles. A particle moves to a new position obtaining a new solution guided by the velocity (a vector). Hence, the velocity plays an important role in affecting the characters of creating

new solution. There are two experience positions used in the PSO; one is the global experience position of all particles, which memorizes the global best solution obtained from all positions (solutions) of all particles; the other is the individual experience position of each particle, which memorizes the local best solution acquired from the positions (solutions) of the corresponding particle has been at. These two experience positions and the inertia weight of the previous velocities are used to determine the impact on the current velocity. The velocity retains part of prior velocity (the inertia) and drives particle toward the direction based on the global experience position and the individual experience position. Thus, the particles can derive new positions (solutions) by their own inertia and experience positions.

Let the search space is D dimension space (the number of dimension is corresponding to the parameters of solutions) and the population consists of Np particles. $X_i^t = \{X_{i1}^t, \dots, X_{iD}^t\}$, X_{iD}^t be the particle i with D dimension space ($i = 1, \dots, Np$) at iteration t. A position X_i^t has a rate of position change called velocity $V_i^t = \{V_{i1}^t, \dots, V_{iD}^t\}$. The individual experience is a position $P_i^t = \{P_{i1}^t, P_{i2}^t, \dots, P_{iD}^t\}$, the local best position for the ith particle until iteration t (called pbest). Additionally, $P_g^t = \{P_{g1}^t, P_{g2}^t, \dots, P_{gD}^t\}$ represents the global best position obtained from among all the population of particles achieved at iteration t (called gbest). The velocity and position of particles could be performed by the Eqs. 1 and 2 in the PSO algorithm.

Where w is an inertia weight used to determine the influence of the previous velocity to the new velocity. The $c1$ and $c2$ are learning factors used to derive how the ith particle approaching the position closes to the individual experience position or global experience position, respectively. Furthermore, the $rand1$ and $rand2$ are the random numbers uniformly distributed in $[0, 1]$, influencing the tradeoff between the global and local exploration abilities during search. A PSO algorithm is described as follows:

1. **Initialization:** Initialize a population of particles with random positions and velocities on D dimensions in the search space.
2. **Update:** Calculate the velocity of each particle by Eq. 3 and move to the next position according to Eq. 4.
3. **Evaluation:** Evaluate the fitness of each particle in the population. Update pbest and gbest positions if necessary, *i.e.*, if the current pbest position is better than the old pbest position, then set the pbest position such that it has the position as the current particle's position. In the same manner, if the current gbest position is better than old gbest position, reset the current gbest position such that it has the position of the current particle.
4. **Termination:** Stop the algorithm if a specified stopping criterion is reached; go to step 2, otherwise.

Proposed PSO for Polarizer Cutting Plan Scheduling Problem (PCPSP)

The most important issue in applying PSO to **PCPSP** is to define an encoding scheme that allows one-to-one mapping between solution and particle (*i.e.*, the particles search solutions in a continuous solution space). This study represents the position of particle i as an p -dimensional vector ($p \hat{=} N$), whose entries' value belongs to the discrete set $N = \{1, 2, \dots, n\}$, where n is the number of knife molds *i.e.*, a particle moves in an p -dimensional search space. The position of

particle i is defined as $X_i^t = (X_{i1}^t, X_{i2}^t, \dots, X_{in}^t)$, $X_{ij}^t \in R$, where X_{ij}^t is the actual position value for knife mold j of particle i at iteration t . For example, a material S02000054 is needed for certain product, and there are 13 knife molds available to be selected for the PCP process. Table 5 represents the mapping between the particle and solution. Table 6 shows the corresponding sequence of each knife mold according to the value of X_{ij}^t . This study utilizes the actual position values of X_{ij}^t to convert continuous position values into a sequence. Notably, the position values in X_{ij}^t do not represent an assigned sequence of each knife mold. Instead, the smallest position value (SPV) rule developed by Bean [11] and Tasgetiren *et al.* [12] is used to convert the actual position values of X_{ij}^t into a processing sequence for each knife mold.

Table 5: Knife Mold to PSO Particle Mapping

Dimension j	knife mold	
0	11R700005R	-0.8
1	51R000001R	3.7
2	11R000001R	1.4
3	11NL00002R	-3.7
4	11N800002R	-1.8
5	51N700002R	-3.8
6	11N000003R	1.2
7	51LI70004R	-2.4
8	51L700003R	1.0
9	11L000004R	1.3
10	11I000005R	2.8
11	118000004R	-0.4
12	117000028R	-3.9

Table 6: The Process Sequence of each Knife Mold

Dimension j	knife mold	X_{ij}	Sequence
12	117000028R	-3.9	1
5	51N700002R	-3.8	2
3	11NL00002R	-3.7	3
7	51LI70004R	-2.4	4
4	11N800002R	-1.8	5
0	11R700005R	-0.8	6
11	118000004R	-0.4	7
8	51L700003R	1.0	8
6	11N000003R	1.2	9
9	11L000004R	1.3	10
2	11R000001R	1.4	11
10	11I000005R	2.8	12
1	51R000001R	3.7	13

```

// Obtain the same material available knife mold
strSQL = "exec usp_PCP_GetBOMNo '{0}','{1}','{2}'";
//Sorted by size
strSQL = string.Format(strSQL, SID, Component, 5);
DataView dv_BOMNo = _t.SQLQuery(strSQL, "");
Rand = new Random();
// Loading available knife mold information
Dimensions = DataviewToArray(dv_BOMNo);
D = Dimensions.Length;
// Define the parameters of PSO
Initial(); for (t = 1; t <= T; t++) {
Update_pBest(); Update_gBest();

    Compute new velocity  $V_{ij}^{new}$  according to Eq. (1)

    Compute new position  $X_{ij}^{new}$  according to  $X_{ij} + V_{ij}$  (Eq. 2)

if (NextGeneration(t))
break; }
// Show the best result

```

Figure 5 illustrates the details of the proposed PSO algorithm for PCPSP.

Computation Results

The PSO approach was implemented in C++ and tested on a Pentium 4 (3.40 GHz). To determine suitable parameter settings, the following parameter values were tested: $c1 = 2$ and $c2 = 2$, the inertia weight w from 0.9 to 0.4 decreased by 0.1. The number of particle is set to 40. Tables 7 and 8 describe the real case for cutting plan of materials S02000053 and S02000054, and the

Table 7: The Information of Product Demand for Material S02000053 and S02000054

Material no.	Product no.	Type of size	Yield rate	Demand
S02000053	FCMF09305301	27"W	0.935	31,947
S02000053	FCMF05805301	23.6"W(M)	0.955	228,599
S02000053	FCPF08205301	21.5"W	0.975	94,048
S02000053	FILF05305301	18.5"W-A	0.965	93,084
S02000053	FCPF15705301	9.0"	0.965	75,956
S02000053	FILF02705301	8"-2	0.965	31,028
S02000053	FHSF01005301	7.0"	0.970	99,429
S02000053	FILF00705301	7.0"	0.970	125,174
S02000053	FHSF10205301	5.0"	0.750	135,911
S02000053	FILF04805302	5.0"	0.750	26,484
S02000054	FCMR09305401	27"W	0.935	25,929
S02000054	FCMR05805401	23.6"W(M)	0.955	204,085
S02000054	FCMR08205401	21.5"W	0.975	124,901
S02000054	FILR05305401	18.5"W	0.965	51,915
S02000054	FCHR02705401	8.0"	0.965	41,875
S02000054	FILR00705401	7.0"	0.970	105,984

Table 8: The Available Knife Molds for Material S02000054

knife mold	Valid Length	Material Width	GAP	Item	size	length	width	No. of pieces	Product no
11R700005R	620.86	1490	7	3	27.0	606.80	345.20	3	FCMR09305401
				4	7.0	157.50	96.70	6	FILR00705401
11R000001R	620.86	1490	7	3	27.0	606.80	345.20	3	DCMR09308801
									FCMR09302601
									FCMR09303201
									FCMR09303202
									FCMR09305401
FCMR09305402									
51R000001R	610.86	1490	7	3	27.0	606.80	345.20	3	FCMR09305401
11NL00002R	1250.08	1490	7	3	23.6	530.22	302.22	6	FCMR05603203
				4	21.5	483.00	275.00	3	FCMR08203202
11N800002R	1288.57	1490	7	3	23.6	530.22	302.22	9	FCMR05605401
				4	8.0	163.60	123.10	6	FCHR02705401
51N700002R	1288.57	1490	7	3	23.6	530.22	302.22	9	FCMR05605401
				4	7.0	157.50	96.70	10	FILR00705401
11N000003R	1288.57	1490	7	3	23.6	530.22	302.22	9	DCMR05606311
									FCMR05602601
									FCMR05603203
									FCMR05605401
									FCMR05605402
									FCMR05605411
									FILR05605401
									FWKR05603201
FWKR05605401									
51LI70004R	1173.09	1490	7	3	21.5	483.00	275.00	6	FCMR08205401
				4	18.5	414.70	235.40	6	FILR05305401
				5	7.0	157.50	96.70	6	FILR00705401
51L700003R	1173.09	1490	7	3	21.5	483.00	275.00	9	FCMR08205401
				4	7.0	157.50	96.70	15	FILR00705401
11L000004R	1173.09	1490	7	3	21.5	483.00	275.00	9	DCMR08208801
									FCMR08202601
									FCMR08202602
									FCMR08203201
									FCMR08203202
									FCMR08205301
									FCMR08205401
									FCMR08205402
FCMR08205411									
11I000005R	1347.11	1490	7	3	18.5	414.70	235.40	16	DILR05302401
									DILR05306301
									DILR05308801
									DILR05309401
									FILR05303801
FILR05305401									
118000004R	528.63	1490	7	3	8.0	163.60	123.10	33	FCHR02700201
									FCHR02701401

Table 9: The Computational Results using Various Approaches for Material S02000054 used

Material no.	Number of knife molds	Experience Automation	Exhaustive System	PSO
S02000054	13	57,506	56,801	56,800
The number of knife molds used		6	6	6
Saving		1,122	1,827	1,828

computational result is represented in table 9 using various approaches for material S02000054 used. **Conclusions**

The proposed PSO method for polarizer cutting scheduling plan can find the near optimal solution, and the cost saving about NT\$1.828 millions for the material S02000054. The traditional experience based method which is based on the bigger polarizer cutting first. In the proposed experience automation, which selects combing the bigger polarizer cutting first and the cutting mold knife with less material waste first, also can save NT\$1.122 millions. And with less cutting can reduce the work force cost. Then the company can produce more polarizer for sale. Finally, using our proposed method not only can reducing the consumption of material cost and work cost, but also can increase the productively.

References

- Reynolds C. Flocks, Herds and Schools (1987) A Distributed Behavioral Model, *Computer Graphics*, 21, 25-34.
- Washer G., Haubner H. and Schumann H (2007) An Improved Typology of Cutting and Packing Problems, *European Journal of Operational Research*, 183, 1109-1130.
- Dyckhoff H. (1990) A Typology of Cutting and Packing Problems, *European Journal of Operational Research*, 44, 145-159.
- Gilmore P. C. and Gomory R. E. (1961) A Linear Programming Approach to the Cutting Stock Problem, *Operations Research*, 9, 849-859.
- Kennedy J and Eberhart RC (1995) Particle Swarm Optimization, *Proceedings IEEE Int'l. Conf. on Neural Networks*, IV, 1942–1948.
- Luo X, Wang D, Tang J and Tu Y. (2006) An Improved PSO Algorithm for Resource-Constrained Project Scheduling Problem, *Intelligent Control and Automation, WCICA 2006, The Sixth World Congress on*, 1, 21-23, 3514–3518.
- Salman A, Ahmad I and Al-Madani S. (2002) Particle Swarm Optimization for Task Assugnment Problem, *Microprocessors and Microsystems*, 26, 363–371.
- Tseng CT and Liao CJ. (2008) A Particle Swarm Optimization Algorithm for Hybrid Flow-Shop Scheduling with Multiprocessor Tasks, *International Journal of Production Research*, 46(17), 4655 – 4670.
- Kuo IH, Horng SJ, Kao TW, Lin TL, Lee CL, Terano T and Pan Y. (2009) An Efficient Flow-Shop Scheduling Algorithm Based on A Hybrid Particle Swarm Optimization Model, *Expert Systems with Applications*, doi:10.1016/j.eswa.2008.08.054, 2009.
- Sha DY and Hsu CY. (2006) A Hybrid Particle Swarm Optimization for Job Shop Scheduling Problem, *Computers and Industrial Engineering*, 51, 791–808.
- Bean JC (1994) Genetics and Random Keys for Sequencing and Optimization, *ORSA J Comput* 6:154–160.
- Tasgetiren MF, Sevkli M, Ling YC and Gencyilmaz G (2004) Particle Swarm Optimization Algorithm for Permutation Flowshop Sequencing Problem, *Lecture Notes in Computer Science*, 3172: 382–390.