An Approach to the Guillotine Strip Packing Problem

Damir Vuk
Visoka škola za menadžment u turizmu i informatici u Virovitici
Trg Ljudevita Patačića 3, 33000 Virovitica, Croatia
damir.vuk@vsmti.hr

Abstract. *A two-dimensional strip packing problem typical for the production of corrugated cardboard is analyzed. This problem is a difficult problem due to a wide variety of parameters and optimizations objectives that have to be accounted for. Specifics of this kind of strip packing problem arise from additional restrictions on standard two-dimensional strip packing problem. Multi-objective optimization in a dynamically changing environment is an additional problem.*

We propose the heuristics based procedure as an acceptable solution to the problem. The main advantage of the approach is possibility of gradual change of requirements and objectives through an evolutionary process of building an acceptable end solution in real time. This is accomplished by the significant reduction of possible solutions search space, and applying this procedure recursively. Evolutionary procedure is based on the possibility of permanent user interaction in each stage of the solution building. We think that this is the major difference between proposed approach and some other models in the literature.

Keywords. Strip packing problem, evolutionary heuristics, corrugator production

1 Introduction

In the last twenty years, corrugated cardboard production changed significantly. Competition is much stronger. Customers orders are becoming smaller, due date shorter. Many of orders are urgent. Orders quantity tolerance is mostly zero. Sales margins on average are weak. Factories are under time and cost pressure. In the past, most factories scheduled orders manually. Nowadays, it is not acceptable.

Then why do many factories still schedule orders manually? As Velasquez et al. stated. [8], in practice some theoretically good solutions are abandoned or not fully used. The major reason is the fact that they do not fully capture the problem complexity. This problem complexity arises from multi objectives that need to be considered.

Corrugated board production is extremely dynamical environment. Lot of different orders, stochastically coming from customers, are to be processed on a daily basis. Hundreds of orders a day, it is not unusual situation, rather usual. The large production equipment, known as corrugator, produce cardboard cutting it by required quantities, in rectangles of different sizes and grades, at the end of cardboard production process. Cardboard can be produced in a few different widths and practically of unlimited length. Because of divergence in the production process accuracy and possible paper side damage, cardboard should be trimmed on each side. Side trim is not strictly fixed. It mainly depends of cardboard grade types produced, but also of actual production circumstances.

The main problem is how to schedule orders with appropriate cutting schema. This is a multi objective problem.

The first objective is to produce all the required orders by required quantities, not more and not less, if possible. If this is not feasible, then some small difference between required and scheduled quantities is acceptable.

The second objective is to design a cutting schema by reducing cardboard waste (trim). Cardboard waste should not exceed desired maximal level, which is directed by internal production and commercial rules. Total waste is expressed as percent of total cardboard area used.

There are more additional objectives and restrictions: all due dates must be satisfied, on the corrugator produced items should be scheduled for second production phase – packaging production, production capacity shall be optimally utilised, etc.

2 The problem definition

This is typical combinatorial problem. It is identified as NP-hard (Rinaldi and Franz) [8]. It differ from some others strip-packing problems. It is a specialization of Open Dimension Problem (ODP) [10] with sequencing constraint as stated in [5] (Velasquez et al.). We will call this type of problem as Strip Packing Problem for Corrugator Production (SPP-CP).

Let we see in more details, what defines a real strip-packing problem in a typical corrugated board plant. Figure 1 shows arrangement of a generic cutting plan (schema). It shows the arrangements of orders in parallel strips of equal items, sequenced as segments on the corrugated cardboard strip.

Because of technical requirements of the corrugator, a typical solution scheme consists of one or more segments. Each segment can include only one or two different orders at the maximum. Each order can be divided into one or more strips of equal lengths and consequently, of equal quantity of items. Total number of strips that segment contains is determined by number of rotating blades, which cuts strips and side trim. Usual number of blades is seven or more. Each strip width is equal and matches to the orders items width.

The corrugator operates with three knives acting as guillotines. First guillotine cuts the incoming cardboard across the whole width. This happens at the beginning, and at the end of segments. If the segment contains two orders, segments are cut up on two longitudinal parts, each containing items of only one order. These segment parts are transported into separate cutters, upper and lower, containing second and third guillotine. The second and the third guillotine (upper/lower) cut each of order strips on appropriate items length.

It is required that at least one of longitudinal segment parts, contains all of the ordered items left. Because of multiple strips, it is not always possible to reach an exact match to the ordered quantity of items. Therefore, it is acceptable nearest value.

Segment length is determined by the longest of his parts. Minimal segment length is restricted on minimal allowable value, and on minimal desirable value. For example: 150m/300m. Minimal desirable value is usually minimal length acceptable. Minimal allowable value is accepted only as exception; if not a better solution is found. Longer segment length value is not restricted; it is welcomed because of higher corrugator's production throughput. Usually, maximal speed of cardboard production is up to 300m per minute, but on shorter segments, it needs to be much lower (about 100m / minute).

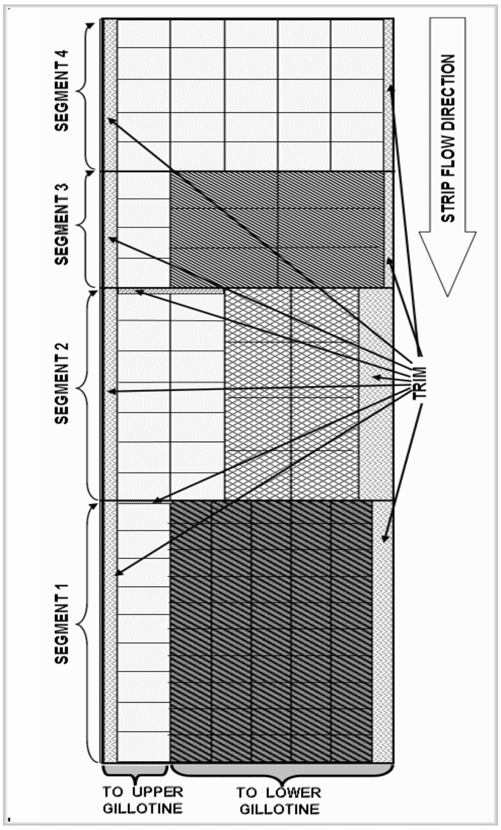

Figure 1. Corrugated cardboard cutting plan schema

Side trim width is defined by technological conditions of the current production process. It can be corrected each time, based on experience by human (user). Some types of items allow zero side trim. Some types of items require being pressed folding scores on it. Total number of folding scores is restricted by type of corrugator (for example 12).

Let we conclude that solution of described problem shall contain one or more cutting plans for each grade-type of order items, and for each production day, with acceptation of given objectives and constraints.

3 Problem data representation

It is highly important to represent problem data on an appropriate way. In literature, there is little evidence how to represent a combinatorial problem data. More emphasis is put on algorithm structure. This is in certain way understandable, because combinatorial problem generally, deals with small amount of input data. This is not the case in the corrugated cardboard production scheduling. Although, here also appear combinatorial problems, they contain a larger amount of data and richer data structure.

It is a natural way to think about data through related sets of data. SPP-CP problem includes a few basic sets of data:

Q - Set of all orders waiting to be scheduled,

O - Set of all orders in process of scheduling and with the same grade-type,

G – Set of all grade-types;

A – Set of all items (articles) in orders ***O;***

P – Solution space, set of all possible segments;

F – Feasible solution space;

T – Set of all candidate segments (target segments);

S – set of pair of orders in a segment of the end solution;

E – Packing plan i.e. set of all segments forming an end solution.

Each of these sets can be further represented as a relational table or relational view. If we do it in this way, we get the possibility for effective transformations of represented data using standard SQL language syntax and especially, good performance inherently embedded in SQL-engine. Before implementation into relational model, it is need to define minimal attribute set for each of data sets.

Q *->(OrderId, ArticleId, Quantity, DueDate, ...)*
O *->(OrderId, QuantityLeft ,Witdh, Length, DueDate, Grade, ScoreNo,...)*
G *->(GradeId, Layers, Wave,..)*
A *->(ArticleId, GradeId, Width, Length, Scores, AtricleType,...)*
P *->(SegmentWidth, SegmentLenght, TrimWidth, Utilisation, OrderA, APlay, OrderB, BPlay, BLenghtLeft,...)*
F *->(SegmentWidth, SegmentLenght, TrimWidth, Utilisation, OrderA, APlay, OrderB, BPlay, BLenghtLeft,...)*
T *->(SegmentWidth, SegmentLenght, TrimWidth, Utilisation, OrderA, APlay, OrderB, BPlay, BLenghtLeft,...)*
E *->(PlanNo, PlanWidth, SegmentNo, SegmentLenght, TrimWidth, Utilisation, OrderA, APlay, OrderB, BPlay, BLenghtLeft,...)*

Is should be noted that sets *Q, G, A* and *E* will be implemented as relational tables, while sets *O, P, F* and *T* will be implemented as relational views. We will later describe relationships between defined sets. Additionally, in a real software implementation, there will be more tables and views, but this is out of scope of this paper.

4 Evolutionary model

Combinatorial problems are characterised by huge solution spaces, and consequently, practical impossibility to search all of items in them. Many solution procedures based on heuristics, are successful in finding good results by reduction of search space, with little or not at all loss of optimality of the end solution.

In this paper, we propose an evolutionary approach as the way for finding solution to SPP-CP. This is not a typical evolutionary algorithm as usually found in the literature [2][3],[4]. It is a kind of “evolutionary breeding” with the goal to find only one - the best individual, from each of new generated population, and then to include that individual as the part of the final solution of the problem.

If we want to model in this way, we shall first define genetic structure of the model. In this model, sets *O, P, F, T, S* and *E* are generalised as populations of individuals with equal genetic structure.

Genetic structure of each individual consists of two chromosomes (***X*** and ***Y***), and a life span value (***L***). Each chromosome consists of chromosome identity (***I***) and of chromosome power (***W***). All properties of an individual arise from his genetic structure. Let C_n be set of properties of an individual n, then we have:

$$C_n=(X_n(I_{x,n},W_{x,n}),\ Y_n(I_{y,n},W_{y,n}),L_n) \qquad (1)$$

Genetic properties of an individual are partially inherited from his parents, trough the crossing, and partly are result of mutations. Chromosome X identity is inherited from one parent, and chromosome Y identity is inherited from another parent. Chromosome's power is not inherited; it is result of a mutation. Life span is result of a genetic calculation; it will be later explained in more detail. There are also individuals with only one ancestor. Those individuals have both chromosome's identities equal, and $W_y=0$:

$$C_n=(X_n(I_{x,n},W_{x,n}),\ Y_n(I_{x,n},,0),L_n) \qquad (2)$$

or:

$$C_n=(X_n(I_{x,n},W_{x,n}),\ X_n(I_{x,n},,0),L_n) \qquad (3)$$

With regard to reproduction, there are basically, two kinds of individuals, i.e. two kinds of populations: base individuals/populations and reared individuals/populations. There is another specific in relation to base individuals, they always are of type (3), and have the chromosome power equal to 1:

$$C_n=(X_n(I_{x,n},1),\ X_n(I_{x,n},\ 0),L_n) \qquad (4)$$

Only base individuals can produce new reared individuals, which are to be reproduced with aims to select the best of them as a candidate for end solution.

Let we see now, what really means the explained genetic representation. Each individual represents the pair of orders, or order alone. Each chromosome identity represents all properties of order. Each chromosome power represents number of strips of the same order. Life span value is calculated differently for each type of individuals. For base individuals L is calculated as

$$L_n= q_n * l_n \qquad (5)$$

where q_n is an remaining quantity of order items (articles), and l_n is length of item n. Remaining quantity is the quantity not included in a packing plan, i.e. this is the difference between ordered quantity and sum of all scheduled quantities.
For reared individuals L is calculated as

$$L_n=\min(\ L_{ax,n}/W_{x,n},,\ L_{ay,n}/W_{y,n}) \qquad (6)$$

where $L_{ax,n}$,and $L_{ay,n}$ are L values of ancestors. Function min() returns lower value of the two expressions, rounded on integer value.

5 Evolutionary process

Evolution process in this model is aimed to generate base populations as basis for the target populations breeding. The base population is the populations of individuals from which a target population is generated. It is selected from new orders queue as a dynamic view with constraints related to end solutions. It contains only individuals (orders) with a life span value greater then zero. Individuals with L=0 are totally scheduled, and shall not to be included in the base population.

From the base population solutions space is to be constructed, which contains all possible combinations with all possible mutations. Solutions space P is defined as a Cartesian product:

$$P=(O\ x\ O\ x\ B\ x\ B\ \ ) \qquad (7)$$

where B is set of numbers from 0 to b:

$$B=(0,\ 1,\ 2,\ 3,...b) \qquad (8)$$

where b is number of corrugator's blades.

Solutions space includes many of unfeasible solutions. Hereafter, the solutions space is to be transformed into feasible solutions space, with accordance to requirements and with accordance to constraints, exposed in the chapter 3.
From the feasible solutions space applying restrictions set by the user (or by default), a target population is to be generated.

Only one the best individual from each new population may be selected as part of end population. What is the best individual in a target population is determined according to fitness function, which is defined by applied selection strategy. There may be more selection strategies in the model, including strategy of manual selection based on user's judgement.

When the best individual is selected, that has two consequences. First, selected individual is to become a part of end solution sequence E. Then, base population Ot, change and become next generation of base population Ot+1. If selected individual originate from only one ancestor, then that ancestor disappear from the new base population.

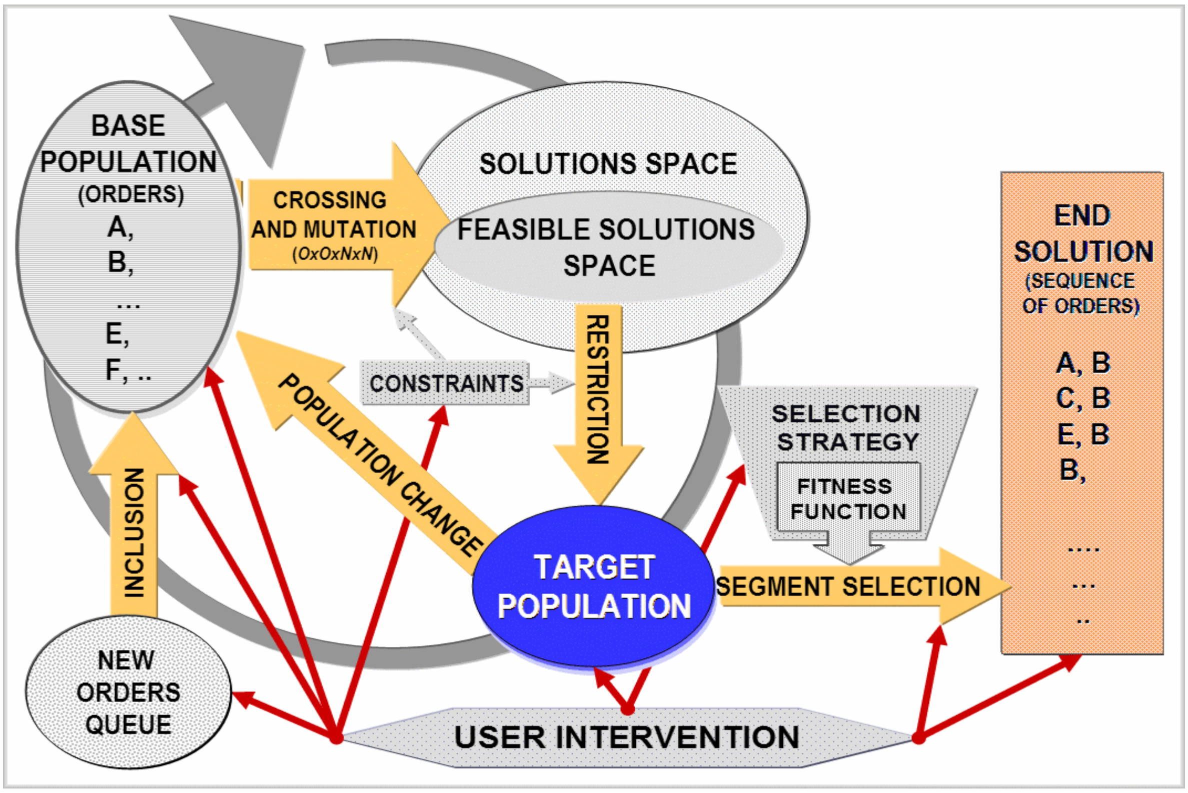

Figure 2. Packing-plan evolution schema

If selected individual originate from two parents, then his live span value is reduced by value of life span and chromosome power engaged in the selected individual. This reduction must be in accordance with eq. 6. As in the first case, base individual whose live span value is reduced to zero or near to zero is to be disappeared.

Evolution process may continue until target population become an empty set. If this happens before an acceptable good end-solution is attained, there are more strategies to apply:

- To change constraints and goals, and to repeat evolution from scratch;
- To remove a bad segment (or more segments) from the end-solution and to repeat evolution from that point;
- To change restrictions of the inclusion and then to repeat the evolution from that point, or from the scratch.

If we apply some of those strategies, the reversible procedure must be triggered, which will bring back all ancestors of the removed segments into a new base population. This is accomplished trough an implementation into relational database. Therefore, all sets defined as views are dynamically updated and all model's data are constantly consistent.

6 Conclusion

The approach to solution of CPP-CP – problem as explained, differ from others approaches and models, especially from the concept of evolution algorithms where a random process of evolution is inherent. Instead, we try to reduce solution space, and by evolution steps search for new solutions.

Here, we included an interactive user intervention in every stage of process of evolutionary solution search. This gives the user a great flexibility, and opportunity to adapting to unpredicted requirements and constraints in the real problem domain. This is in accordance with notice of Wiers in [11]: "*The human scheduler will remain an indispensable factor in the scheduling process. However, many techniques do not account for interaction with the human scheduler.*"

The model allows inherently, applying multi objective strategies. It also allows use of different selection strategies, including selection based on user's judgement. It is open to implement new ones. This openness is a direction for future improvement of the model.

The model is implemented as a software solution, and tested in the real environment. The first user impressions highlight the efficacy and the flexibility of use [9].

References

[1] Doerner K. F, Gendreau M. Greistorfer F, Gutjahr W, Hartl R and Reimann M: **Metaheuristics - Progress in Complex Systems Optimization**, Springer Science+Business Media, LLC, 233 Spring Street, New York, NY, 10013, USA, 2007.

[2] Michalewicz Z: **Genetic Algorithms + Data Structures = Evolution Programs**, Springer-Verlag, Berlin Heidelberg 1996,

[3] Michalewicz Z., Schmidt, M., Michalewicz M. and C. Chiriac : **Adaptive Business Intelligence**, Springer-Verlag, Berlin Heidelberg 2007,

[4] Michalewicz Z., Schmidt, M., Michalewicz M. and C. Chiriac: Case Study: **An Intelligent Decision - Support System**, IEEE Intelligent Systems, Vol.20, No.4. July/Aug 2005,

[5] Rinaldi F, Franz A: **A two-dimensional strip cutting problem with sequencing constraint,** European Journal of Operational Research 183 (2007), Elsevier, 2007, pp. 1371-1384, Available online 7 July 2006 at www.sciencedirect.com

[6] Tyugu E: **Algorithms and Architecture of Artificial Intelligence**, IOS Press, Farifax, VA, USA, 2007.

[7] Valenzula C.L, Wang P.Y: **Heuristics for Large Strip Packing Problems with Guillotine Patterns: an Empirical Study**, MIC'2001 – Metaheuristics International Conference, Porto, Portugal, July 16-20, 2001.

[8] Velasquez D. G. A, Bellini G, and Paternina-Arboleda C. D: **A multi-objective approach based on soft computing techniques for production scheduling in Corrugator manufacturing plants**, Ingenieria & Desarrollo, Universidad del Norte, No. 21, Jun, 2007., ISSN: 0122-3461, pp. 73-92.

[9] Vuk D: **OptiVAL – inteligentni sustav za podršku u donošenju odluka o proizvodnji**, CASE 20 – Metode i alati za razvoj poslovnih informatičkih sustava, Opatija, Croatia, Jun 3-5, 2008.

[10] Wäscher G, Haußner H, Schumann H: **An improved typology of cutting and packing problems,** European Journal of Operational Research 183 (2007), Elsevier, 2007, pp. 1109–1130, Available online: 19 June 2006 at www.sciencedirect.com

[11] Wiers V. C. S: **Human.computer interaction in production scheduling - Analysis and design of decision support systems for production scheduling,** PhD-Thesis, Technische Universiteit Eindhoven, Eindhoven, 1997.