Genetic Algorithm for Optimization of Angle Bar Inventory for Lattice Towers Used in Distribution and Transmission Systems.

Gregory Levitin, Senior Member IEEE; Shmuel Mazal-Tov; Hanoch Ben-Haim

Reliability Department, Planning, Development & Technology Division, Israel Electric Corporation Ltd., Haifa, Israel

Abstract

The problem of achieving maximal economy when angle bars for an electric lattice tower are made from the purchased fixed length blanks is considered. The two-stage optimization tool is proposed. At the first stage, this tool searches for the optimal blank cutting that allows all the necessary bars to be made from the minimal number of blanks by minimizing waste material. At the second stage, the optimal blank purchasing schedule is found which minimizes the total inventory cost in accordance with a preliminary defined plan of lattice tower construction.

1. Introduction

The main building elements of lattice towers used in distribution and transmission systems are angle bars of different lengths and profiles. These bars are made by cutting a blanks of standard length which are purchased by the electric utility in accordance with its need. The proper cutting of these blanks into angle bars allows the total number of blanks to be minimized, which leads to the total lattice tower cost reduction.

After such an optimization, the cost of blank orders corresponding to different number of lattice towers can be a function of this number that grows more slowly than a linear function. Therefore, the larger the order, the cheaper may be the single lattice tower cost. On the other hand, if the order is greater than the need in the lattice towers for the given time, the redundant angle bars should be stored untill the time that, according to the plan, additional lattice towers will be built. The tradeoff exists between the storage cost and the economical savings achieved by increasing the order size. Therefore, a compromising optimal solution which minimizes the total inventory cost (sum of storage costs and purchasing costs) during the given planning horizon can be found. The shipping costs are neglected in this paper which is justified by the fact that the electric lattice tower blanks are usually not shipped separately and, therefore, it is difficult to estimate their specific portion in the total shipping cost as a function of their quantity.

The planning period is considered in this work as a sequence of stages during which the numbers of lattice towers to be built are defined. The optimal blank purchasing is considered to be a combination of two separate optimization problems: (I) optimal blank cutting and (II) determination of the optimal blank purchasing schedule. A general optimization approach is used for the both problems, namely a Genetic Algorithm (GA). The basic GA procedures are adapted for solving the problems.

As the intervals between different stages are short enough, the interest rate effects are not considered in this paper.

2. Problem formulation

According to its specification, each type of lattice tower consists of a given number of angle bars of different lengths and profiles. The total number of different profiles is P. For each profile p ($1 \le p \le P$), a list $\{l_{pi}\}$ ($1 \le i \le M_p$) is specified. This list defines lengths of bars which should be supplied. Here M_p is the total number of bars with profile p. The entire set of angle bars the lattice tower consists of is referred to as complete bar set (CBS).

The angle bars are cut off from blanks that have fixed length L. To produce N lattice towers (N CBSs) the blank set $\mathbf{Q}(N)=\{Q_p(N),\ 1 \le p \le P\}$ should be purchased where $Q_p(N)$ is the number of blanks of profile p necessary to cut the required angle bars of this profile. The entire purchasing cost for $\mathbf{Q}(N)$ is

$$C(N) = \sum_{p=1}^{P} c_p Q_p(N), \qquad (1)$$

where c_p is the cost of single blank of profile p.

The costs C(N) can be minimized by solving the optimal blank cutting problem. This problem is: how to group the bars to be cut from the blanks so that all the required bars of the given profile would be made using the minimal number of blanks. For each p ($1 \le p \le P$) the problem can be formulated as follows:

$$\mathbf{X} = \arg\{Q_p(N, \mathbf{X}) \to \min\}$$

$$\text{subject to}$$

$$\sum_{j=1}^{Q_p} x_{ij} = 1 \quad \text{for} \quad 1 \le i \le M_p, \qquad (2)$$

$$\sum_{i=1}^{M_p} x_{ij} l_{pi} \le L \quad \text{for} \quad 1 \le j \le Q_p,$$

where $\mathbf{X}$ is a matrix of decision variables. For each $x_{ij} \in \mathbf{X}$: $x_{ij}=1$ if bar i is cut from blank j; otherwise $x_{ij}=0$. The formulated problem is widely known as one-dimensional bin packing problem [1]. It is proven that the problem is NP-hard. Several algorithms are suggested for solving this problem[1,5-10].

Note that rather frequently $Q_p(N)<N\cdot Q_p(1)$ because the increase in the required number of bars provides more flexibility for optimal cutting. In the general case, the optimal blank set purchasing cost C(N) is a nonlinear function of order size N. Hence, it may be profitable to distribute the blank set orders in such a manner which allows the total inventory cost to be minimized. This distribution should meet the demand at each stage of planning horizon.

Assume demand D_i to be given for each stage i ($1\leq i\leq K$). If the order size at each stage is equal to the demand, the total inventory cost is

$$C_{tot} = \sum_{i=1}^{K} C(D_i).$$

If at the first stage the order size corresponds to $N_1>D_1$ CBSs, the inventory cost at this stage is $C(N_1)+s\Delta_1$, where $\Delta_1=N_1-D_1$ is the remainder equal to number of CBSs that should be stored after termination of stage 1and s is the cost of storage required for one CBS during one stage.

At the second stage, the minimal order size that can satisfy the demand corresponds to $M_2=\max(0,D_2-\Delta_1)$ CBSs. The remainder after ordering set $\mathbf{Q}(N_2)$ is

$$N_2-(D_2-\Delta_1)=N_1-D_1+N_2-D_2$$

CBSs and the total inventory cost for two stages is

$$C(N_1)+C(N_2)+s(N_1-D_1+N_2-D_2)$$

etc. Assuming that at K-th stage the order size corresponds to $\max(0,D_K-\Delta_{K-1})$ CBSs, we obtain the total inventory cost for arbitrary feasible order distribution $\mathbf{N}=\{N_j\,|\,1\leq j\leq K\}$:

$$C_{tot} = \sum_{i=1}^{K} C(N_i) + s\sum_{i=1}^{K-1}\sum_{j=1}^{i} (N_j - D_j),$$

where solution feasibility condition is:

$$N_i \geq M_i = \max(0, D_i - \Delta_{i-1}) = \max(0, D_i - \sum_{j=1}^{i-1}(N_j - D_j)), \quad 1 \leq i < K. \tag{3}$$

The total inventory cost can also be expressed as

$$C_{tot} = \sum_{i=1}^{K} C(N_i) + s\sum_{i=1}^{K-1}(K-i)\cdot(N_i - D_i). \tag{4}$$

To see that order distribution which is not equal to demand distribution can lead to the total cost reduction, let us estimate the effect of moving one CBS from stage m order to an earlier order at stage m-1. (Initial distribution **N** is assumed to be equal to the demand distribution). It can be easily seen from (4), that purchasing cost will change by

$$C(N_{m-1}+1)+C(N_m-1)-C(N_{m-1})-C(N_m).$$

On the other hand, one more CBS should be stored at stage m-1 which increases the total storage cost by s. The total cost variation caused by the change in order distribution is

$$C(N_{m-1}+1)+C(N_m-1)-C(N_{m-1})-C(N_m)+s.$$

If the total cost decreases, the change is justified. Therefore, the condition of moving one blank set from stage m order to stage m-1 order is

$$s < C(N_{m-1})-C(N_{m-1}+1)+C(N_m)-C(N_m-1).$$

The objective of the order optimization problem is to find the order distribution **N** which minimizes the total inventory cost:

$$\mathbf{N}=\arg\{C_{tot}(\mathbf{N},s)\rightarrow\min\}. \tag{5}$$

3. Optimization technique

A two-stage optimization approach is suggested in this paper. In the first stage, the optimal cutting procedure groups the required angle bars of each profile in such a way that allows the required number of CBSs to be obtained from the minimal number of blanks. This procedure solves the problem (2) for each profile and for different order sizes N and evaluates corresponding minimal purchasing costs C(N) using (1). In the second stage, the minimal cost order distribution is obtained for the given demand distribution by solving optimization problem (5) with costs obtained in the first stage.

The same optimization technique is used for solving both optimization problems. This technique is Genetic Algorithm inspired by a principle of evolution. A brief introduction to Genetic Algorithms is presented in [2]. More detailed information on GA can be found in Goldberg's comprehensive book [3], and the recent developments in GA theory and practice can be found in Ref. [4]. The GA was applied for solving bin-packing problem in a number of works [5-10].

Unlike various constructive optimization algorithms which use sophisticated methods to obtain a good single solution, the GA deals with a set of solutions (population) and tends to manipulate each solution in the simplest way. "Chromosomal" representation requires the solution to be coded as a finite length string. The basic structure of the version of GA used in this paper, which is referred to as GENITOR [11], is as follows.

First, the initial population of N_S randomly constructed solutions (strings) is generated. Within this population new solutions are obtained during the genetic cycle using crossover and mutation operators. Crossover produces a new solution (offspring) from a randomly selected pair of parent solutions providing inheritance of some basic properties of the parents in the offspring.

Mutation results in slight changes in the offspring structure and maintains diversity of solutions. This procedure avoids premature convergence to a local optimum and facilitates jumps in the solution space.

Each new solution is decoded and its objective function (fitness) values are estimated. These values, which are a measure of quality, are used to compare different solutions.

The comparison is accomplished by a selection procedure that decides which solution is better: the newly obtained one or the worst solution in the population. The better solution joins the population and the worse one is discarded. If the population contains equivalent solutions following selection, redundancies are eliminated and, as a result, the population size decreases.

After new solutions are produced N_{rep} times, new randomly constructed solutions are generated to replenish the shrunken population, and a new genetic cycle is started.

The GA is terminated after N_c genetic cycles do not attain improvement of the best-in-population solution. The final population contains the best solution achieved. It also contains different near optimal solutions which may be of interest in the decision making process.

To apply the genetic algorithm to a specific problem, one has to define the solution representation as well as the corresponding procedures. Since the two problems that are to be solved by the GA have different nature, the solution encoding technique and corresponding crossover, mutation and solution fitness evaluation procedures should also differ. The crossover operation should create a feasible solution as offspring of a pair of existing ones. The mutation should also provide feasibility of solutions obtained by random changes in a string representing a solution. The following is a description of the encoding technique and basic procedures for the optimization problems considered in this work.

3.1. Solution representation and GA basic procedures for Problem I.

In this problem of blank cutting, a solution for profile p is represented by M_p length string of integer numbers which represent different angle bars. The order in which the numbers appear in the string determines their grouping within different blanks. Each number should appear in the string only once. The following procedure determines blank cutting based on the order of bar numbers in the string:

Step 1. Set Y=0, k=1, R=0, H=0. Where Y is current total length of angle bars cut off from the same blank, k is current number of bars considered, R is the total length of remainder and H is the total number of blanks used.

Step 2. If $k \le M_p$, perform Step 3; else stop the procedure.

Step 3. Look for first string element g_i ($k \le i \le M_p$) such that $Y+l(g_i) \le L$. If such an element exists perform step 4; else perform step 5.

Step 4. Set $Y=Y+l(g_i)$. Swap the elements in positions k and i and increment k by 1. Return to Step 2.

Step 5. Find element g_j ($k-1 \le j \le M_p$) such that $l(g_j)-l(g_{k-1})$=max while $Y+l(g_j)-l(g_{k-1}) \le L$. Set $Y=Y+l(g_j)-l(g_{k-1})$. Swap the elements in positions k-1 and j. Set R=R+L-Y, H=H+1, Y=0. Return to Step 2.

After the procedure termination, H is equal to the total number of blanks needed ($H=Q_p$) and R is equal to the total remainder length. In order to let the GA look for the solution with minimal number of blanks or, equivalently, with minimal total remainder length, the solution quality (fitness) is evaluated as R.

Solution feasibility means that the solution string contains all the bar numbers of the given profile, and each number appears in the string only once. Any omission or duplication of numbers constitutes an error. A crossover procedure that provides offspring feasibility which was first suggested in [12] and was proven to be highly efficient [13] is used in this work. This procedure first copies all the string elements from the first parent to the same positions of the offspring. Then all the offspring elements belonging to the fragment, defined as a set of adjacent positions between two randomly defined sites, are reallocated within this fragment in the order they appear in the second parent. The following is an example of the crossover procedure in which thc fragment is marked by bold font.

First parent: $g_1\ g_2\ \mathbf{g_3\ g_4\ g_5\ g_6\ g_7}\ g_8\ g_9\ g_{10}$
Second parent: $g_7\ g_8\ \mathbf{g_9\ g_2\ g_4\ g_5\ g_1}\ g_3\ g_6\ g_{10}$
Offspring: $g_1\ g_2\ \mathbf{g_7\ g_4\ g_5\ g_3\ g_6}\ g_8\ g_9\ g_{10}$.

The mutation procedure used in our GA just swaps elements initially located in two randomly chosen positions of the string. This procedure also preserves solution feasibility.

3.2. Solution representation and GA basic procedures for Problem II.

In this problem each element g_i of K-1 length integer string represents the number of redundant CBSs that should be made at stage i in addition to the minimal possible number of CBSs. The solution string, therefore, should contain K-1 integer numbers in the range $0 \le g_i \le G$, where G is the preliminary specified limit.

For each stage i ($1<i<K$) the minimal possible order size M_i is calculated using (3) (note that $M_1=D_1$). The actual order size at stage i determined by the solution string is defined as $N_i=M_i+g_i$. At the last stage, the order size N_K is equal to M_K. For the given $N=\{N_i \mid 0 \le N_i \le K\}$ the total inventory cost (solution fitness) is evaluated using expression (4).

The same crossover technique as used in [14] is adopted in this work. In this technique the fragment of the string is randomly chosen as a set of adjacent positions. All the elements allocated within the fragment are copied onto the child solution string from its first parent, and the rest of the elements are copied from the second one as in the following example.

First parent: $\mathbf{g_1\ g_2\ g_3\ g_4\ g_5\ g_6\ g_7\ g_8\ g_9\ g_{10}}$
Second parent: $g_1\ g_2\ g_3\ g_4\ g_5\ g_6\ g_7\ g_8\ g_9\ g_{10}$
Offspring: $\mathbf{g_1\ g_2}\ g_3\ g_4\ g_5\ g_6\ g_7\ \mathbf{g_8\ g_9\ g_{10}}$

The mutation procedure used is the same as for Problem I.

3.3. Computational effort.

The C language realization of the algorithm on a DEC station 5000/240 is currently in use in the Israel Electric Corporation. The time required to obtain optimal blank cutting for 12 CBSs consisting of 300 angle bars by GA with population size 100 and 300 genetic cycles, each containing 2000 crossovers, does not exceed 15 min. GA with the same parameters solves the optimal blank purchasing schedule generation problem for 20 stages within 80 seconds.

4. Illustrative example

The lattice tower with CBS presented in Table 1 is made of 128 angle bars of 6 different profiles. The bars can be cut off from the blanks with length 9000 mm. The costs of blanks of different profiles are also presented in Table 1.

Table 1. CBS for a lattice tower

Profile	Cost per 1 m	Angle bars required
1	6.30	8*3060; 4*2422; 8*1194
2	7.80	8*2334; 2*1410; 4*1280; 8*1168; 4*1030
3	7.10	8*2791; 8*2256; 8*1660; 2*1516; 8*1380; 4*812
4	8.43	4*2256; 8*1670; 4*1036; 4*582
5	9.10	4*3000
6	5.13	4*2670; 12*1125; 4*775

In the first stage of optimization, the optimal blanks cutting problem was solved for different order sizes (the example of optimal cutting of 8 blanks to obtain angle bars of profile 3 for one CBS is presented in Table 2).

Table 2. Example of profile 3 blank cutting for single lattice tower

N	Cutting bars					Remainder
1	2791	2256	1516	1516	812	109
2	2256	2256	1660	1380	1380	68
3	2791	2256	1380	1380	812	381
4	2791	1660	1660	1380	1380	129
5	2791	2256	1660	1380	812	101
6	2791	2256	1660	1380	812	101
7	2791	2256	2256	1660	-	37
8	2791	2791	1660	1660	-	98

The minimal cost order sets with corresponding **Q**(N) and C(N) for $1 \le N \le 12$ are presented in Table 3. This table also contains the remainder indices α which are equal to percentage of unused material. For each order size N and profile p:

$$\alpha = Q_p \cdot L - \sum_{i=1}^{M_p} l_{pi} /(Q_p \cdot L).$$

One can see that this index has tendency to decrease with growth of N, which causes a reduction in cost per CBS. The comparison between functions C(N) and N·C(1) is presented in Fig. 1, which illustrates how the savings achieved by the optimal blank cutting grow as N increases.

The solutions of inventory optimization problem for an 18-stage period are presented in Table 4. This table contains demand in CBSs for each stage, the cost of solution without CBS storage (N=D in each stage) and costs for two optimal solutions obtained for different storage costs. The order size distributions for these optimal solutions are presented, as well as the number of CBSs stored at each stage.

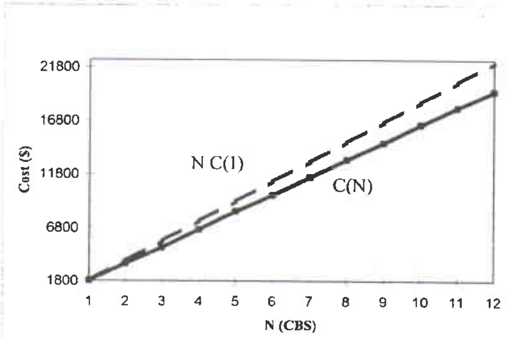

Figure 1: Order cost as function of required No of CBSs.

5. Conclusion

To minimize the purchasing and inventory cost of angle bars used to build electric lattice towers, the two-stage optimization algorithm is suggested. At the first stage, the optimal blank cutting problem is solved for each type (profile) of the bars in order to minimize the total material waste for different number of CBSs ordered. At the second stage, the optimal blank purchasing schedule is developed with respect to lattice tower building plane and to bars storage cost. The total inventory cost is minimized in this stage. Both problems have similar combinatorial nature and, therefore, the same optimization technique is used to solve them. This technique is the biologically-inspired genetic algorithm which imitates the evolution process. The basic GA procedures are developed to fit the algorithm to the specific problems. The algorithm is currently in use in the Israel Electric Corporation.

Table 3. Costs and contents of different order sizes

Order size (CBS)	No of Blanks in Order												Order Cost ($)
	Type of Profile												
	1		2		3		4		5		6		
N	Q_1	α	Q_2	α	Q_3	α	Q_4	α	Q_5	α	Q_6	α	**C(N)**
1	6	19.04	5	10.94	8	1.42	4	19.84	2	33.33	4	24.22	**1854.36**
2	11	11.68	10	10.94	16	1.42	7	8.39	3	11.11	7	13.40	**3448.08**
3	16	8.92	14	4.58	24	1.42	10	3.81	4	0.00	10	9.07	**4971.60**
4	21	7.47	19	6.25	32	1.42	13	1.35	6	11.11	13	6.74	**6647.22**
5	26	6.58	24	7.23	40	1.42	17	5.70	7	4.76	16	5.28	**8316.81**
6	31	5.98	28	4.58	48	1.42	20	3.81	8	0.00	19	4.28	**9840.33**
7	36	5.54	33	5.54	56	1.42	23	2.42	10	6.67	22	3.56	**11515.95**
8	42	7.47	38	6.25	64	1.42	26	1.35	11	3.03	25	3.00	**13166.37**
9	47	6.98	42	4.58	72	1.42	30	3.81	12	0.00	28	2.57	**14765.76**
10	52	6.58	47	5.26	80	1.42	33	2.84	14	4.76	31	2.22	**16441.38**
11	57	6.25	52	5.80	88	1.42	36	2.03	15	2.22	34	1.93	**18035.10**
12	62	5.98	56	4.58	96	1.42	39	1.35	16	0.00	37	1.69	**19558.62**

Table 4. Optimal order distribution.

	C_{tot}		Stages																	
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Demand	121971.77		**10**	**1**	**5**	**7**	**1**	**2**	**2**	**8**	**1**	**12**	**2**	**3**	**1**	**2**	**1**	**9**	**5**	**1**
Optimal for s=39.4	120243.96	ordered	12	0	6	6	0	4	0	9	0	12	3	3	0	3	0	9	6	0
		stored	2	1	2	1	0	2	0	1	0	0	1	1	0	1	0	0	1	0
Optimal for s=21.2	119955.83	ordered	12	0	6	6	0	4	0	9	0	12	6	0	0	3	0	12	3	0
		stored	2	1	2	1	0	2	0	1	0	0	4	1	0	1	0	3	1	0

References

[1]. S. Martello and P. Toth, *Knapsack problems*. Wiley, Chichester, 1990.

[2]. S. Austin, An introduction to genetic algorithms. *AI Expert*, 1990, **5**, 49-53.

[3]. D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, Reading, MA, 1989.

[4]. T. Bäck, *Evolutionary Algorithms in Theory and Practice. Evolution Strategies. Evolutionary Programming*. Genetic Algorithms. Oxford University Press, 1996.

[5]. C. R. Reeves, A genetic approach to bin-packing, *In Proc. of the Second Finnish Workshop on Genetic Algorithms and their Applications*, Vaasa, Finland (1994) 35-49.

[6]. Cheng-Yan Kao and Feng-Tse Lin, A stochastic approach for the one-dimensional bin-packing problems, *In Proc. of the 1992 IEEE Int. Conf. on Systems, Man, and Cybernetics*, vol. 2, (1992) 1545-1551.

[7]. S. Khuri, M. Schutz, J. Heitkotter, Evolutionary heuristics for the bin-packing problem, *In Proc. of the ICANNGA'95*, Ales, France (1995) 285-288.

[8]. D. Smith, Bin packing with adaptive search, *In Proc. of the First Int. Conf on Genetic Algorithms and their Applications*, Pittsburgh (1985) 202-207.

[9]. E. Falkenauer, *Genetic Algorithms and Grouping Problems*, John Wiley & Sons., 1998.

[10]. E. Falkenauer and A. Delchambre, A genetic algorithm for bin packing and line balancing, *In Proc. of the 1992 IEEE Int. Conf. on Robotics and Automation*, Vol. 2, Nice, France (1992) 1186-1192.

[11]. D. Whitley and J. Kauth, GENITOR: a different Genetic Algorithm, *Tech. Rep. CS-88-101*, Colorado State University, Fort Collins, Co., 1988.

[12]. J. Rubinovitz and G. Levitin, Genetic algorithm for assembly line balancing, *Int. Journal of Production Economics*, 42 (1995) 343-354.

[13]. G. Levitin, S. Masal-Tov, D. Elmakis, Genetic algorithm for open loop distribution system design, *Electr. Power Syst. Res.*, 32 (1995) 81-87.

[14]. A. Lisnianski, G. Levitin, H. Ben-Haim, D. Elmakis, Power System structure optimization subject to reliability constraints, *Electr. Power Syst. Res.*, 39 (1996) 145-152.