OPTIMAL DISTRIBUTION OF SUB-ASSEMBLIES IN STORES OF FACTORY BY EVOLUTIONARY ALGORITHMS

Bogna MRÓWCZYŃSKA
Faculty of Transport, Silesian University of Technology,
Krasińskiego 8, 40-019 Katowice, e-mail: bogna.mrowczynska@polsl.pl

Summary

The paper deals with an application of evolutionary algorithms for optimisation of sub-assemblies distribution in the stores of factory. Numerical model is presented. The fitness function is expressed as a function of distances between stores and assembly rooms and costs of inner transport. Penalty function is used to include restrictions. The results showed that the applied method is the efficient tool for solving such problems.

Keywords: evolutionary algorithm, optimisation, store.

OPTYMALIZACJA ROZKŁADU PODZESPOŁÓW W MAGAZYNACH FABRYKI PRZY POMOCY ALGORYTMÓW EWOLUCYJNYCH

Streszczenie

W artykule przedstawiono zastosowanie algorytmów ewolucyjnych do optymalizacji rozłożenia podzespołów i materiałów wykorzystywanych w produkcji w magazynach zakładu produkcyjnego. Przedstawiono model numeryczny problemu. Funkcję przystosowania wyrażono jako funkcję odległości pomiędzy magazynami a halami montażowymi i kosztów wewnętrznego transportu między nimi. Ograniczenia na pojemność poszczególnych magazynów uwzględniono stosując funkcję kary. Otrzymane wyniki są optymalne i potwierdzają skuteczność algorytmów ewolucyjnych w rozwiązywaniu tego typu problemów.

Słowa kluczowe: algorytmy ewolucyjne, optymalizacja, magazyn.

1. INTRODUCTION

Storage is one of the most important link of the logistic chain [4]. Functional efficiency of the storage depends on the storage systems [3]. Optimisation of sub-assemblies distribution in stores of factory is one of such elements. The proper allocation of the sub-assemblies influences the costs of inner transport. There are real money paid for fuel, amortisation and costs of employment of operators of fork-lift trucks. As usual, no optimisation is applied.

This paper presents an application of evolutionary algorithm for optimisation of sub-assemblies distribution in the stores of factory.

2. NUMERICAL MODEL

The factory consist of eight assembly rooms and seventeen stores (fig. 1). The problem is, how to place all sub-assemblies in stores to minimise costs of inner transport used to carry sub-assemblies from store to assembly rooms. Every sub-assembly should be put in one store, but in every store can place a few sub-assemblies. Every store has its capacity. The volume of all sub-assemblies in one store couldn't overstep its capacity.

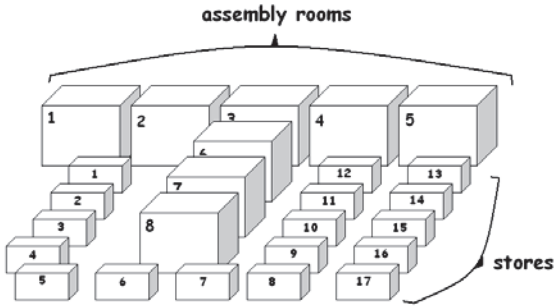

Fig. 1. The factory with assembly rooms and stores

Thus the problem of optimisation consist in finding minimum of the following function:

$$f = \sum_{j=1}^{m} \sum_{k=1}^{l} \sum_{i=n_j} C_{jk} \cdot d_{ik} \cdot l_{jk} \qquad (1)$$

where

m - number of the sub-assemblies,

l - number of the assembly rooms,

n_j - the ordinal number of the store, in which is placed the i-th sub-assembly

c_{jk} - the costs of transport one unit of the j-th sub-assembly to the k-th assembly room (here $c_{jk} = 1$),

d_{ik} - the length of the road from the i-th store to k-th assembly room (table 1),

Table 1. The length of the road from the i-th store to k-th assembly room

length of the road [m]		assembly rooms							
		1	2	3	4	5	6	7	8
stores	1	92,1557	102,6025	57,8305	134,316	21,6398	98,8715	89,1709	53,7264
	2	86,1861	96,6329	51,8609	128,3464	15,6702	92,9019	83,2013	47,7568
	3	80,2165	90,6633	45,8913	122,3768	9,7006	86,9323	77,2317	41,7872
	4	74,2469	84,6937	39,9217	116,4072	14,5509	80,9627	71,2621	35,8176
	5	68,2773	78,7241	33,9521	111,1838	20,5205	74,9931	65,6656	30,2211
	6	74,2469	72,7545	27,9825	117,1534	26,4901	69,0235	71,6352	36,1907
	7	80,2165	66,7849	22,0129	123,123	32,4597	63,0539	77,6048	42,1603
	8	86,1861	60,8153	16,0433	129,0926	38,4293	57,0843	83,5744	48,1299
	9	82,8282	54,8457	10,0737	135,0622	44,3989	51,1147	89,544	54,0995
	10	76,8586	48,8761	10,0737	141,0318	50,3685	45,1451	95,5136	60,0691
	11	70,889	42,9065	16,0433	147,0014	56,3381	39,1755	101,4832	66,0387
	12	64,9194	36,9369	22,0129	152,971	62,3077	33,2059	107,4528	72,0083
	13	65,6656	37,6831	27,9825	152,2248	68,2773	27,2363	111,5569	77,9779
	14	71,6352	43,6527	33,9521	146,2552	74,2469	21,2667	117,5265	83,9475
	15	77,6048	49,6223	39,9217	140,2856	80,2165	15,2971	123,4961	89,9171
	16	83,9475	55,965	46,2644	134,316	86,1861	9,3275	129,4657	95,8867
	17	91,4095	63,427	53,7264	128,3464	92,1557	16,7895	135,4353	101,8563

l_{jk} - number of unith j-th sub-assembly transfer to the the k-th assembly room (table 2).

The constrains are described as follows:

$$h = \sum_{i=1}^{n} w_i \sum_{j=1}^{m} (V_{ij} - V_{gi}) \qquad (2)$$

where

V_{ij} - volume of the j-th sub-assembly in the i-th store,

V_{gi} - capacity of the i-th store,

w_i - penalty coefficient.

Table 2. The number of the unit j-th sub-assembly, which were transferred to the k-th assembly room

unit per week		assembly rooms							
		1	2	3	4	5	6	7	8
sub-assemblies	1	51	85	102	0	68	85	0	68
	2	51	0	102	119	0	0	0	34
	3	68	85	0	119	68	68	0	68
	4	0	0	0	0	0	0	51	51
	5	51	0	0	0	68	0	0	51
	6	0	0	102	0	0	68	0	0
	7	17	51	0	0	68	0	0	17
	8	0	17	0	0	17	0	0	17
	9	0	0	0	0	0	51	0	0

If $\sum_{j=1}^{m} (V_{ij} - V_{gi}) > 0$ then the i-th store is overloaded and $w_i > 0$.

If $\sum_{j=1}^{m} (V_{ij} - V_{gi}) \le 0$ then the i-th store is not overloaded and $w_i = 0$.

Thus the optimisation problem consist in finding minimum of the following function:

$$\tilde{f} = \sum_{j=1}^{m} \sum_{k=1}^{l} \sum_{i=n_j} C_{jk} \cdot d_{ik} \cdot l_{jk} + \\ + \sum_{i=1}^{n} w_i \sum_{j=1}^{m} (V_{ij} - V_{gi}) \qquad (3)$$

3. EVOLUTIONARY OPTIMISATION

The problem of optimisation of the sub-assemblies distribution in the stores of the factory is solved by an evolutionary algorithm. It is one of the methods of artificial intelligence. The evolutionary algorithm is inspired by natural evolution. The algorithm of the evolutionary optimisation is presented in Fig. 2. The algorithm has been described in details elsewhere [1, 2].

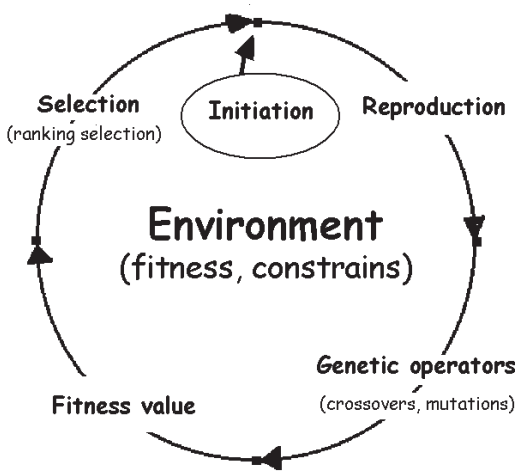

Fig. 2. An algorithm of evolutionary optimisation

Environment

The problem is formulated as follows:

The k-th chromosome from the population, which represents the possible solution is expressed in the form:

$$\mathbf{n_k} = <n_{k1}, n_{k2}, \ldots, n_{k7}> \quad (4)$$

where:

n_{ki} - i-th gene in the k-th chromosome. It is an ordinal number of the store, in which is placed the i-th sub-assembly.

The population is a set of chromosomes. They will evolve during all process of the artificial evolution.

Fitness function

Because the fitness function should be maximised, it is expressed as follow:

$$\hat{f} = \frac{1}{\tilde{f}} \quad (5)$$

If the value of the fitness function $\hat{f}$ is higher then the solution is better. The optimal solution is obtained by maximising the fitness function with respect to $\mathbf{n_k}$.

Genetic operators

This evolutionary algorithm uses operators of a crossover (non-uniform crossover, arithmetic crossover, heuristic crossover) and a mutation (uniform mutation, boundary mutation, non-uniform mutation) described in [2]. The genetic operators work with the assumed probabilities. Several number of the tests were calculated in order to tune their proper running.

Selection

This evolutionary algorithm uses operators of the proportional selection in order to select better solutions. Elitist model is used to remember best solution [2].

4. RESULTS

The maximal size of the chromosome population used in the evolutionary computation was 100. The maximal length of the life was 20000. The calculation were repeated 100 times for every set of parameters. The probabilities of the evolutionary operators, the penalty coefficients and the results of the example calculations are presented in tables 3.

The value of the fitness function of the best solution is equal 0,9139. Almost all results fulfil all assumptions and constrains. The constrains were overloaded in the variant 2 and 3, but just this ones on every 100 repeated calculations. It happened for penalty coefficient equal 10. For less penalty coefficient it didn't happen. Thus the results are promising.

The diagram 3 presents the changes of the values of the fitness function trough all generations.

5. CONCLUSIONS

The paper presents the application of evolutionary algorithms for optimisation of the sub-assemblies distribution in the store of the factory to minimise costs of inner transport. The best solutions obtained during the evolutionary optimisation have the highest fitness value and they fulfils all the constrains. The worst results appeared infrequently only for the high value of the penalty coefficient.

The numerical model is simply. Thus the evolutionary algorithm is an efficient tool for solving such discrete optimisation problems.

Table 3. Results of evolutionary optimisation

Variant	Penalty coefficient	Probability of operators:						Value of the best fitness function	
		Non-uniform crossove	Arithmetical crossove	Heuristic crossove	Uniform mutation	Boundary mutation	Non-uniform mutation		
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	overload *10* [%]
1	15	0,9	0,9	0,9	0,1	0,1	0,1	0,8792	0
2	10	0,7	0,7	0,7	0,3	0,3	0,3	0,8763	20,094
3	10	0,8	0,8	0,8	0,2	0,2	0,2	0,8922	20,094
4	5	0,9	0,9	0,9	0,1	0,1	0,1	0,9139	0
5	5	0,8	0,8	0,8	0,2	0,2	0,2	0,9114	0
6	5	0,9	0,8	0,9	0,2	0,1	0,1	0,9131	0
7	5	0,8	0,9	0,9	0,2	0,2	0,1	0,9131	0
8	5	0,9	0,9	0,5	0,2	0,1	0,1	0,9139	0
9	5	0,9	0,8	0,7	0,1	0,2	0,3	0,9030	0
10	5	0,7	0,7	0,7	0,3	0,3	0,3	0,8761	0
11	5	0,9	0,8	0,7	0,3	0,2	0,1	0,9139	0

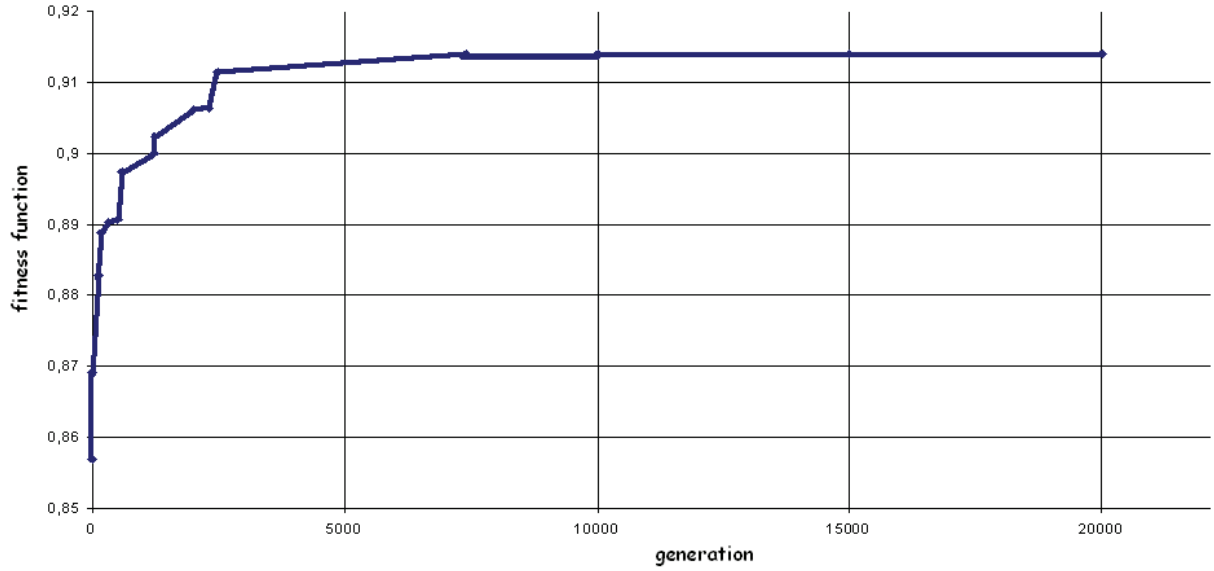

Fig. 3. The diagram of values of fitness function in the best variant

REFERENCES

[1] Arabas J.: *Wykłady z algorytmów ewolucyjnych.* Wydawnictwo Naukowo – Techniczne, Warszawa 2001.

[2] Michalewicz Z., *Genetic Algorithms + data Structures = Evolutionary Programs.* Springer-Verlag, Berlin 1996.

[3] Mrówczyńska B.: *Optimal goods distribution in supermarket's store by evolutionary algorithms.* AI-METH Series, Gliwice, 2007.

[4] Skowronek C., Sariusz – Wolski Z.: *Logistyka w przedsiębiorstwie.* PWE, Warszawa 2003.