UDK 004.421:519.87:658.524:005.591.1

PLANIRANJE SERIJSKE PROIZVODNJE UPORABOM GENETSKOG ALGORITMA
PLANNING OF A SERIAL PRODUCTION BY MEANS OF GENETIC ALGORITHMS

Domagoj CRLJENKO - Dalibor BLAŽEVIĆ - Tonči MIKAC

Sažetak: *U ovom članku predstavljen je koncept za optimalno planiranje serijske proizvodnje uporabom genetskog algoritma i njegova aplikacija u vidu kompjutorskog programa. Prikazani evolucijski program u kratkom vremenu pronalazi optimalno rješenje, što omogućuje brzo donošenje poslovnih odluka i na taj način poboljšava konkurentnost poduzeća.*

Ključne riječi:
- genetski algoritam
- serijska proizvodnja
- optimizacija
- UOBX križanje

Summary: *The aim of this work is to present a concept for the optimal planning of serial production by means of a genetic algorithm and its application by way of a computer program. The displayed evolutional program in short time finds an optimal solution, ensuring fast business decision making and thereby enhancing the competitiveness of the enterprise.*

Key words:
- genetic algorithm
- serial production
- optimization
- UOBX crossover

1. UVOD

U mnogim poduzećima planiranje proizvodnjom predstavlja velik problem [1]. Takva poduzeća karakterizira loša usluga prema kupcima u smislu nepoštivanja rokova isporuke, povećanoga obujma zaliha, neadekvatne uporabe proizvodne opreme i radne snage, niskih koeficijenata obrtaja financijskih sredstava vezanih za materijal i velikoga broja radnog osoblja namijenjenog rješavanju gorućih proizvodnih problema [2].

2. DEFINICIJA PROBLEMA

Zadatak je u proizvodnom pogonu optimizirati planiranje proizvodnje, i to upotrebom genetskog algoritma. Konkretno, u zadanim okvirima proizvodnih kapaciteta potrebno je optimizirati vremena proizvodnje za određenu količinu proizvodnih naloga.

Genetski algoritam treba u svojim kromosomima (jedinkama) sadržavati ukupne radne zadatke za proizvodne kapacitete u vremenskom intervalu, vodeći računa o serijskoj proizvodnji (slijedu događaja), te o

1. INTRODUCTION

Production planning of the manufacturing process presents an arduous problem for many enterprises [1]. Such corporate enterprises are characterized by inadequate service toward customers with respect to delivery terms, increased supplies volume, inadequate utilization of manufacturing equipment and labor force, low turnover coefficient of financial resources connected to material and a large number of personnel assigned to solve the most acute of production problems [2].

2. THE PROBLEM DEFINITION

The aim is to optimize production planning by means of a genetic algorithm. Explicitly, it is necessary to optimize manufacturing time within a given scope of manufacturing facilities and for a particular amount of manufacturing orders.

A genetic algorithm has to include in its chromosomes (individuals) overall production tasks for manufacturing facilities in a given time interval, while taking account of the serial production (consecution of tasks), and the dostupnosti pojedinih proizvodnih kapaciteta u određenom vremenskom intervalu.

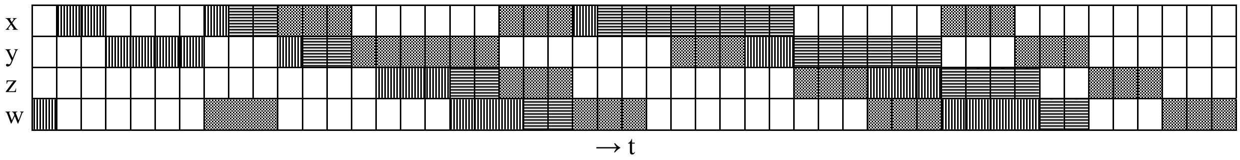
availability of particular manufacturing facilities in that given time interval.

3. FORMULACIJA PROBLEMA

Kromosom će biti višedimenzionalan. Osnovna će dimenzija biti vrijeme, dok će ostale dimenzije predstavljati proizvodni procesi, proizvodni kapaciteti, radni nalozi, nivoi (razine) iz sastavnica, te redni brojevi operacija iz tehnoloških postupaka.
Osnovna struktura takva kromosoma prikazana je na slici 1.:

3. THE PROBLEM FORMULATION

The chromosome will be multidimensional. The basic dimension will be time, whereas the other dimensions should represent manufacturing processes, manufacturing capacity, work orders, bill of material levels (layers), and ordinals of operations from the technological procedure.
The main structure of such a chromosome is presented in Figure 1.



Slika 1. Struktura kromosoma
Figure 1. Chromosome structure

- ▥ – prvi radni nalog
- ▤ – drugi radni nalog
- ▩ – treći radni nalog
- x – tehnološka operacija x
- y – tehnološka operacija y
- z – tehnološka operacija z
- w – tehnološka operacija w

- ▥ – first work order
- ▤ – second work order
- ▩ – third work order
- x – technological operation x
- y – technological operation y
- z – technological operation z
- w – technological operation w

Slika 1. prikazuje trajanje (x–os) za četiri tehnološke operacije (x, y, z i w) te tri radna naloga.
Takvu strukturu potrebno je proširiti uključivanjem razina (nivoa) iz sastavnica gotovih proizvoda i poluproizvoda, te rednim brojem operacija iz tehnoloških postupaka.
Struktura tako proširenoga kromosoma može se opisati pseudokodom:

Figure 1 shows the duration (x–axis) for four technological operations (x, y, z and w) and three labor orders.
It is necessary to broaden such a structure by including layers (levels) from the bill of materials of finished and semi-finished products, and ordinals from the technological process.
The structure of such an extended chromosome can be described by the pseudo-code:

```
Structure Chromosome()
{
        Resources[]      As Resource
        Fitness          As Single
}
Structure Resource()
{
        Code             As String
        Genes[]          As OneGene
        Duration         As Integer
}
Structure OneGene()
{
        Order_No         As String
        Layer            As Integer
        Operation_No     As Integer
}
```

Slika 2. Pseudokod strukture kromosoma
Figure 2. Pseudo-code of a chromosome structure

Kromosom će dakle sadržavati u svojoj strukturi niz procesa (Resources []), te pripadajuću vrijednost dobrote jedinke Fitness, koju će izračunavati unutar evolucijskog procesa.
Svaki proces predstavlja moguću operaciju koja se može izvesti u proizvodnom pogonu. Općenito je zamišljeno da se u jednom procesu kromosoma nalazi jedan stroj, ili jedan radnik, ili jedna tehnološka operacija. Ako proizvodni pogon posjeduje više jednakih strojeva (*kapacitet*), kromosom će sadržavati više procesa s jednakim kodom (*Code*). Jedan proces sadržava i ukupno vrijeme trajanja procesa (*Duration*), jer moguće je pretpostaviti da proces neće biti u potpunosti iskorišten.
Svaki se proces sastoji još i od niza gena (Genes []).
Jedan se gen sastoji od broja odnosno (Order_No) šifre naloga, razine (*Layer*) sklopa, koji predstavlja razinu sastavnice gotovog proizvoda, te redni broj operacije iz tehnološkog postupka (Operation_No).
Funkcija dobrote (*fitness*) sastoji se u određivanju kvalitete pojedine jedinke [3].
Budući da će svaka jedinka sadržavati sve podatke, i to: naloge, količine naloga, proizvodne kapacitete, sastavnice i tehnološke postupke, funkcija dobrote sastoji se u pretraživanju prostora populacije za najboljom jedinkom. To znači da će najbolja jedinka biti ona kojoj je za izvršavanje zadanih radnih naloga na zadanim proizvodnim kapacitetima potrebno najmanje vremena. Pojednostavljeno, najbolja će jedinka biti ona kojoj će duljina gena u svim pojedinačnim procesima biti najmanja.
Odnosno:

Therefore, the chromosome in its structure will include a series of processes (Resources []), and a particular individual fitness value, which will be calculated within the evolution process.
Each process presents a possible operation that can be carried out in the manufacturing plant. It is generally conceived that one process of the chromosome consists of one machine, one worker, or one technological operation. If a manufacturing plant contains several machines of equal characteristics (*capacity*), the chromosome will contain several processes with an equal code (*Code*). A single process also contains the overall time of process duration (*Duration*), since it can be assumed that the process will not be completely exploited.
Each process is also comprised of a series of genes (Genes []).
One gene is composed of the number i.e., the code number (Order_No) of the order, the level (*Layer*) of the product assembly (which represents the constituent level (*threshold*) of the finished product), and the ordinal of the operation from the technological procedure (Operation_No).
The effectiveness of the fitness function lies in the quality of individuals [3].
Since each individual will contain all data information, namely: orders, order quantities, manufacturing capacities, bill of material and technological processes, the fitness function consists of a search for the best individual in a population range. That means that the best individual will be the one that needs the least amount of time to execute all the given labor orders for a given resource. Simply, the best individual will be the one that has the shortest gene length in all single processes.
That is:

$$f = \frac{\sum t_{nal}}{t_{\max}} \cdot K_{fitness} \qquad (1)$$

pri čemu je:
- $\sum t_{nal}$ – zbroj trajanja svih naloga,
- $t_{\max}$ – trajanje najdužeg procesa jedinke,
- $K_{fitness}$ – koeficijent translacije

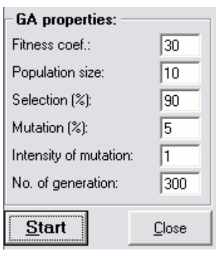
where:
- $\sum t_{nal}$ – duration sum of all orders,
- $t_{\max}$ – duration of largest individual process,
- $K_{fitness}$ – translation coefficient

4. RAZRADA GENETSKOG ALGORITMA

4. 1. Kontrolni parametri genetskog algoritma

4. GENETIC ALGORITHM ELABORATION

4. 1. Genetic algorithm control parameters



Slika 3. Parametri evolucijskog programa
Figure 3. Evolutional program properties

Zbog lakšeg korištenja, te mogućnosti bržeg pronalaska dobre jedinke, u programsku je aplikaciju postavljena mogućnost mijenjanja parametara evolucijskog programa (slika 3.)

Due to easier usage and the ability of quickly finding a good individual, the option of changing evolutional program parameters has been implemented in the program application (Figure 3.)

4. 2. Inicijalizacija populacije

Inicijalizacija populacije, općenito, predstavlja kreiranje jedinki proizvoljnih (slučajnih) karakteristika [4]. Budući da se proizvodnja odvija serijski, u samom procesu inicijalizacije potrebno je poslužiti se determinističkom metodom smještaja pojedinih tehnoloških operacija u strukturu jedinke.
Na taj će se način na samom početku evolucijskog procesa dobiti dovoljno dobre jedinke nad kojima će genetski operatori djelovati.

4. 2. Population initialization

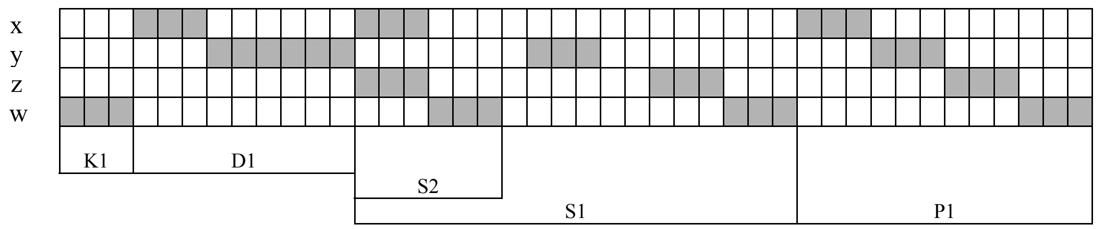
Population initialization generally represents the creation of individuals with arbitrary (random) characteristics [4]. Since manufacturing is carried out serially, it is necessary in the process of initialization to accomplish a deterministic method for setting a particular technological operation into the individual structure.
This approach ensures, from the outset of an evolutional process, a satisfactory solution that will yield adequately efficient individuals over which genetic operators will perform.

4. 3. Prikaz radnog naloga pseudonalozima

Radni nalog opisuje što se proizvodi i u kolikim količinama. Tehnološkim postupcima opisano je koje se tehnološke operacije koriste, u kojem obujmu i kojim redoslijedom djeluju na gotov proizvod. Pritom gotov se proizvod može sastojati od drugih poluproizvoda ili čak gotovih proizvoda, koji također imaju svoje tehnološke postupke [5].
Radi pojednostavnjenja algoritma, smanjenja potrebe i utjecaja determinističke metode smještaja naloga u jedinku, radni je nalog podijeljen prema nizu tehnoloških postupaka potrebnih po pojedinoj razini proizvodnje. Tako dobiveni dijelovi ukupnoga radnog naloga nazvani su pseudonalozima.
Npr. radni nalog sastoji se od izrade 3 komada gotovog proizvoda P1. Proizvod P1 sastoji se od podsklopova S1 i S2. Za izradu sklopa S1 potrebno je proizvesti dio D1. Prije početka proizvodnje dijela D1 potrebno je izvršiti kontrolno mjerenje K1 (slika 4.).

4. 3. Presenting work order using pseudo-order

The work order describes what is manufactured, and in what quantities. The technological processes describe the required technological operations, in what amount, and in which order of precedence they affect the final product. A final product can thus consist of other semi-final products, or even final products that have their own technological processes [5].
Due to a simplification of the algorithm, a reduction of the purpose and the effect of the deterministic method on the setting of orders in the individual, the work order is divided according to a series of technological processes required in a certain level of manufacturing. Parts of the overall work order obtained in such a way are called "pseudo-orders".
E.g., a work order consists of producing three pieces of the final product P1. The final product P1 consists of the subsets S1 and S2. For the production of the subset S1, it is necessary to produce the subset D1. Before producing the subset D1 it is necessary to make a control-measuring K1 (Fig. 4).



Slika 4. Podjela radnog naloga na pseudonaloge
Figure 4. Division of a work order into pseudo-orders

Radni nalog sa slike 4. sastojat će se od 5 pseudonaloga, i to: K1, D1, S1, S2 i P1. Pseudonalozi su međusobno ovisni o smještaju, npr. pseudonalog K1 mora biti smješten ispred pseudonaloga D1, pseudonalog D1 mora biti smješten prije početka proizvodnje sklopovlja S1 i S2. Pseudonalozi S1 i S2 u jednom se vremenskom dijelu proizvode zajedno, odnosno istodobno, jer nam to

The work order from Figure 4 will consist of five pseudo-orders, namely K1, D1, S1, S2 and P1. Between them the pseudo-orders depend on location, e.g., the pseudo-order K1 must be located ahead of the pseudo-order D1; the pseudo-order D1 must be located before the beginning of the manufacturing subsets S1 and S2. The pseudo-orders S1 and S2 are both manufactured over one time interval tehnološki postupci omogućuju, a sve to prije početka izrade gotovog proizvoda P1.

(because the technological processes allows for this) and all of the above is to happen prior to the outset of the manufacturing of the final product P1.

4. 4. Pseudokod evolucijskog programa

Na slici 5. vidljiv je pseudokod evolucijskog programa.

4. 4. Evolution program pseudo-code

Figure 5 shows the pseudo-code of the evolution program.

```
Genetic_algorithm ()
{
        Population_Initialization(i=0)
        i = 1;
        Do While i < No_Generations
        {
                i = i + 1;
                Fitness ()
                Sorting ()
                Selection (% Selection)
                Crossover ()
                Mutation_Chromosome (% Mutation)
                Mutation_PseudoN (% Mutation)
        }
        Solution_decision ()
}
```

Slika 5. Pseudokod evolucijskog programa
Figure 5. Evolution program pseudo-code

4. 5. Selekcija

U svakom evolucijskom ciklusu svi kromosomi prolaze kroz eliminacijski proces. Nakon što se za svaku jedinku odredi dobrota (*fitness*), odabire se definirani postotak (*% Selection*) dobrih jedinki koje se prenose u sljedeću generaciju. Manjak jedinki nastao procesom eliminacijske selekcije nadomješta se križanjem jedinki.

4. 5. Selection

In each evolutional cycle, all chromosomes go through an elimination process. After the determination of a fitness level for every individual, a defined percentage (*% Selection*) of good individuals are selected, which are propagated in the subsequent generation. A lack of individuals caused by the elimination process is resolved by a crossover of individuals.

4. 6. Križanje

Za proces križanja kromosoma razvijen je model križanja zasnivan na UOBX (*Uniform Order Based Crossover*) metodi [6].
Naime, nemoguće je odrediti gene u kromosomima za križanje klasičnim metodama, jer metoda smještaja naloga, pseudonaloga i pojedinih tehnoloških operacija zahtijeva upotrebu determinističkih metoda. To znači da bi križanje dvaju kromosoma na nekom proizvoljnom genu, proizvelo jedinke potpuno neupotrebljive, ili bi vjerovatnost nastanka dobre jedinke bila vrlo mala. Na taj bi se način evolucijski program učinio neupotrebljivim.
UOBX metoda zasniva se na proizvoljnoj definiciji maske (uniforme) i redoslijedu smještaja pseudonaloga u jedinku. Na osnovi maske odbacuju se ili čuvaju pseudonalozi u dvjema jedinkama (roditeljima). Nakon toga prazna se mjesta nadopunju pseudonalogom iz drugog roditelja, a koji još nije unesen u kromosom.
Npr. neka je redoslijed pseudonaloga u kromosomima:

4. 6. Crossover

For the crossover of chromosomes, a new model of crossover method was developed based on the UOBX method (Uniform Order Based Crossover) [6].
In fact, it is impossible to assign genes in chromosomes for a crossover using classical methods, because a method of setting orders, pseudo-orders and particular technological operations requires the use of deterministic methods. That means that a crossover of two chromosomes on some randomly chosen gene will create unserviceable individuals, or the chance of originating an optimal individual will be very small. In that way, the evolutional program would be unserviceable.
The UOBX method is based on the random definition of the mask (uniform), and on the relative order of the pseudo-orders in an individual. Based on the mask in parent individuals, pseudo-orders are discarded or retained. After that, an empty position is filled by a pseudo-order from another parent that has not yet been implemented in the chromosome.

E.g., Allowing that the order of pseudo-orders in the chromosome is:

$Chromosome_1$: 5-3-2-1-0-4
$Chromosome_2$: 0-4-2-5-1-3

Kreira se maska slučajnim odabirom:

The mask is created randomly:

Mask: 1-0-1-0-1-0

Iz prvog se kromosoma oduzimaju pseudonalozi, koji na pripadajućem mjestu u masci imaju 0, a kod drugog kromosoma 1:

Pseudo-orders with the flag 0 in a specific place within the mask are excluded from the first chromosome, and in the other chromosome pseudo-orders with the flag 1 are excluded:

$Chromosome_{1x}$: 5-_-2-_-0-_
$Chromosome_{2x}$: _-4-_-5-_-3

Zatim se kreira novi kromosom na način da se prazni pseudonalozi nadopunjuju preostalim pseudonalozima iz druge jedinke, ili onim pseudonalozima koji nedostaju, te se dobivaju nove jedinke:

The new chromosome is created in such a way that empty pseudo-orders are filled with the remaining pseudo-orders from other individuals or with newly created pseudo-orders, and in this way new individuals will emerge:

$NewChromosome_1$: 5-4-2-3-0-1
$NewChromosome_2$: 2-4-0-5-1-3

Na ovaj se način omogućuje križanje jedinki, točnije rečeno, pseudonaloga unutar kromosoma, a što u stvari predstavlja križanje jedinki i njihovih stvojstava.

The crossover of individuals is thereby enabled, or more precisely stated, the pseudo-orders of chromosomes, and this actually represents a crossover of individuals and their properties.

4. 7. Mutacija

Mutacija je općenito proces u kojem se jedinki promijeni slučajno odabrani gen [7].
Ali, slično kao i kod križanja, pojedini gen nije moguće izmijeniti slučajno, a da se pritom ne naruši redoslijed tehnoloških operacija i pseudonaloga.
Mutacija kao operator mora osigurati populaciji stalni priljev svježih (drugačijih) jedinki kako populacija ne bi završila u lokalnom optimumu, koji ne mora nužno biti globalni optimum. Mutiranje je zato u ovom modelu izvedeno na način da se izvršavaju dva slična operatora mutiranja, pri čemu jedan djeluje nad pseudonalogom, a drugi nad cijelom jedinkom.
Prva metoda iz jedinke obriše pseudonalog, te ju ponovno pokuša smjestiti, dok druga metoda generira jedinku na isti način kako se u početku generira cijela populacija.
Mutacija će, izvedena takvom metodom, osigurati stalno pretraživanje cijelog prostora mogućih rješenja.

4. 7. Mutation

Usually, mutation is a process of changing one randomly chosen gene of an individual [7].
However, as in a crossover, it is not possible to change some of the genes randomly and not to consequently violate the precedence of technological operations and the sequence of pseudo-orders.
The mutation as a genetic operator must guarantee that the population of individuals is constantly replenished with fresh (different) individuals, assuring in this way that the characteristics of a population would not finish in a local optimum that is not necessarily a global optimum. Because of that, the mutation in this model is performed in such way that two similar genetic operators are executed during the mutation, in the aim that the first will act upon a pseudo-order, and other will act upon the whole individual.
The first method deletes a pseudo-order from an individual, and tries to allocate it again, and the second method generates an individual in an identical way as observed during population initialization.
The mutation, performed in this way, will provide a permanent searching over a whole range of possible solutions.

5. PRAKTIČNA MJERENJA

Mjerenja su izvršena na desktop-računalu sljedećih karakteristika:
- OS: Windows XP Pro
- CPU: Pentium IV 1,6 GHz
- 512 MB DDR RAM.

Općenito se pokazalo da genetski algoritam daje slabe rezultate pri finom podešavanju parametara. Bez obzira na to provedena su preliminarna mjerenja ulaznih parametara genetskog algoritma kako bi se vidjelo s kojim ulaznim parametrima uopće započeti program. Rezulati su pokazali sljedeće:
- veličina populacije povećava i ubrzava vjerojatnost pronalaska rješenja,
- veći postotak selekcije osigurava raznolikije, ali i slabije jedinke u križanju, dok će manjim postotkom selekcije izumrijeti više lošijih jedinki. Brže izumiranje jedinki može algoritam odvesti u lokalni minimum, pa je potrebno smanjenjem postotka selekcije povećati postotak mutiranja,
- postotak mutiranja ne bi trebao biti prevelik, jer može kreirati lošije jedinke od svojih prethodnika, pa bi se takvim velikim postotkom algoritam samo nepotrebno ponavljao. Također mutacija mora stvarati nove, drugačije jedinke da bi osigurala algoritmu križanje nove i drugačije jedinke,
- broj generacija je proizvoljan. Veći broj osigurava pronalazak prihvatljivog rješenja, ali i produljuje trajanje programa.

Za potrebe mjerenja napravljeni su zamišljeni tehnološki postupci i sastavnice proizvoda.
Svi podaci unose se šifrirani, a evolucijski program u svom radu koristi šifre iz unosa.
Za tehnološke operacije u program su unijeti sljedeći podaci (tablica 1.):

5. PRACTICAL MEASUREMENTS

Measurements are executed on a desktop computer with the following characteristics:
- OS: Windows XP Pro
- CPU: Pentium IV 1,6 GHz
- 512 MB DDR RAM

Generally, it was demonstrated that the genetic algorithm gives inferior solutions when applied with a fine-tuning of parameters. Regardless of that, the preliminary measurements of input parameters of the genetic algorithm were performed in order to decide which parameters to start the program with. The results were evidence of the following:
- the size of a population increases and accelerates the possibility of finding a solution,
- a higher percentage of selection secures diversity but a weaker quality of individuals in the crossover, however with a lower percentage of selection a significant number of individuals will dye. Faster dying of individuals may bring the algorithm to a local minimum, so that it is necessary to lower the selection percentage to increase mutation percentage,
- the percentage of mutation should not be too high because the creation of bad individuals from its parents could occur, so with that kind of high percentage the algorithm would needlessly repeat. Likewise, the mutation must produce new and different individuals in order to ensure the algorithm new and different individuals for crossover,
- the number of the generation is random. A higher number ensures the discovery of an acceptable solution, but extends the program duration.

For measuring purposes, an imaginary technological process and bill of material of products have been made.
All data are input in code-letter form, and the code-letters from the input are used in the evolutional program.
Data for technological operations in the program are represented in Figure 1:

Tablica 1. Tehnološke operacije – resursi
Table 1. Technological operations – resources

Oznaka / Code-letter	Tehnološka operacija / Technological operation	Kapacitet / Capacity
Tok	Tokarenje / Turning	1
Glod	Glodanje / Milling	2
Buš	Bušenje / Drilling	2
Skla	Sklapanje / Assembling	2

Također, upotrijebljeni su sljedeći tehnološki postupci za pojedine dijelove (proizvode) (tablica 2.):

In addition, the following technological process is used for particular parts (products) (Fig. 2):

Tablica 2. Tehnološki postupci
Table 2. Technological process

Šifra dijela / Ident code	Tehnološka operacija / Technological operation	Trajanje / Duration
0001	Tok Glod Buš Skla	1 min. 1 min. 1 min. 1 min.
0002	Buš Skla	1 min. 1 min.
0003	Buš Skla	1 min. 2 min.
5001	Tok Glod Buš Skla	1 min. 2 min. 3 min. 4 min.
5002	Tok Glod Buš Skla	4 min. 3 min. 2 min. 1 min.
5003	Tok Glod Buš Skla	1 min. 1 min. 1 min. 1 min.
5004	Tok Glod Buš Skla	3 min. 2 min. 3 min. 1 min.
5005	Tok Glod Buš Skla	1 min. 1 min. 1 min. 1 min.

Upotrijebljene su i zamišljene sastavnice dijelova (tablica 3.):

Imaginary bills of material of products (parts) are used (Table 3):

Tablica 3. Sastavnice dijelova
Table 3. Bill of material of products

Šifra dijela / Ident code	Sastavnice dijelova / Bill of material of products
0002	0003 – 1 kom./piece
5001	0001 – 1 kom./piece 0002 – 2 kom./pieces
5002	0001 – 2 kom. /pieces 0003 – 1 kom. /piece
5003	0001 – 1 kom. /piece 0002 – 1 kom. /piece
5004	0001 – 1 kom. /piece 0003 – 1 kom. /piece
5005	0001 – 1 kom. /piece 0002 – 1 kom. /piece

Planirana je proizvodnja sljedećih dijelova (tablica 4.):

Manufacturing of the following products is planned (Table 4):

Tablica 4. Planirani radni nalozi
Table 4. Planned labor-orders

Šifra dijela / Ident code	Količina / Quantity
5001	1 kom. /piece
5002	2 kom. /pieces
5003	3 kom. /pieces
5004	2 kom. /pieces
5005	1 kom. /piece

Upotrijebljeni su sljedeći ulazni parametri genetskog algoritma (tablica 5.):

The following input parameters of the genetic algorithm are used (Table 5):

Tablica 5. Ulazni parametri genetskog algoritma
Table 5. Input parameters of genetic algorithm

Fitness coefficient	30
Population size	10
Selection	90 %
Mutation	5 %
Intensity of mutation	1
No. of generations	300

Rezultat evolucijskog programa dan je na slici 6.

Evolucijski je program dobio prikazani optimalni rezultat nakon 30 sekundi.
Fitness najbolje jedinke je 52, dok je za izvršenje proizvodnje zadane tablicama 1 – 4, predloženo trajanje od 46 minuta.

The results of the evolutional program are given in Figure 6.

The evolutional program achieved the optimal solution after 30 seconds, as presented.
The fitness of the best individual is 52, whereas for the manufacturing execution given in Tables 1-4, the recommended execution time is 46 minutes.

6. ZAKLJUČAK

Ovaj rad pokazuje da se upotrebom genetskog algoritma planiranje serijskog modela proizvodnje može učiniti lakšim i jednostavnijim, a ponajprije ekonomičnijim.
Razvijeni model, koji se zasniva na upotrebi genetskog algoritma, ali i upotrebom determinističke metode, pokazao se kao adekvatan alat prilikom optimiziranja vremena izvršavanja pojedinih funkcija.
Mjerenja su pokazala da algoritam u relativno kratkom vremenu pronalazi dobro rješenje, te je stoga pouzdan za korištenje, a može se i višestruko ponavljati.

6. CONCLUSION

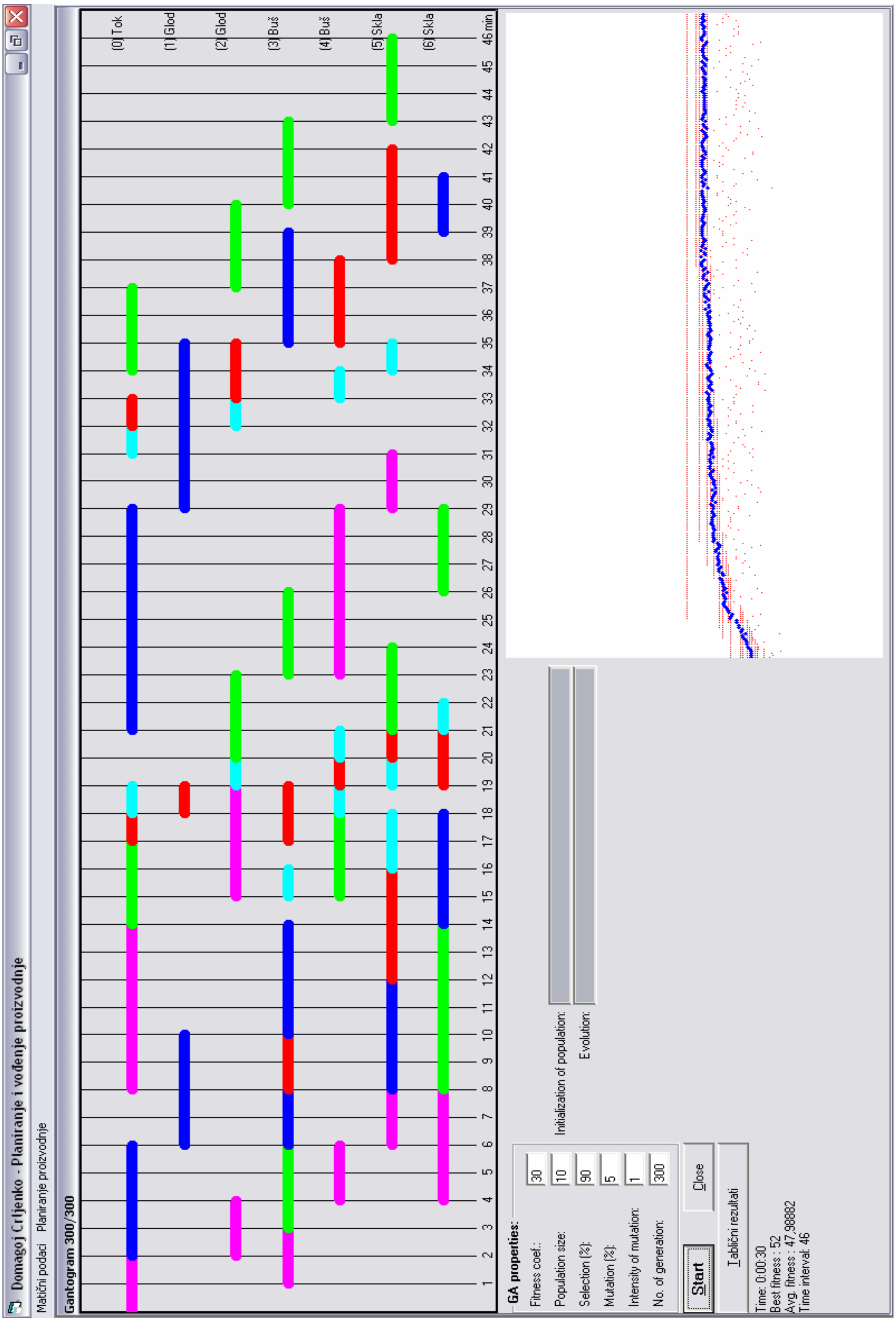
This paper demonstrates that the usage of a genetic algorithm in production planning based on a serial model can be performed simpler and more easily, and notably more economically.
The developed model, based on the usage of the genetic algorithm, and with the usage of the deterministic method, proved to be an adequate tool for time optimization in the execution of certain functions.
Measurements showed that the algorithm finds out a satisfactory solution in a relatively short time span, thereby it is reliable for using, and can be multiplied repeatedly.

Obavijest

Članak je rezultat istraživanja provedenih na projektu provođenog uz potporu Ministarstva znanosti, obrazovanja i športa Republike Hrvatske, modeliranje naprednih proizvodnih struktura kod inteligentne proizvodnje 069-0692976-1740.

Acknowledgement

The presented results are derived from the scientific research project ‘Modeling of Advanced Production Structures of Intelligent Manufacturing’, No. 069-0692976-1740, with the support of the Croatian Ministry of Science, Education and Sports.



Slika 6. Evolucijski program
Figure 6. Evolutional program

LITERATURA
REFERENCES

[1] Mikac, T.: *Organizacija i upravljanje proizvodnjom*, Tehnički fakultet Rijeka, Rijeka, 2004.
[2] Blažević, D.: *Softverska podrška aktivnostima plansko-izvršne funkcije poduzeća*, Tehnički fakultet Rijeka, Rijeka, 2005.
[3] Golub, M.: *Genetski algoritam*, Fakultet elektrotehnike i računarstva Zagreb, Zagreb, 2004.
[4] Haupt R. L. & S. E.: *Practical Genetic Algorithms*, New York, 1998.
[5] Perinić, M; Mikac, T. & Ljubetić, J.: *Applying genetic algorithm in activities of process plan selection*, The 12th International DAAAM Symposium "Intelligent Manufacturing & Automation: Focus on Precision Engineering", Jena, 2001,
[6] H. Lee and S.-S. Kim: *Integration of process planning and scheduling using simulation based genetic algorithm*, The International Journal of Advanced Manufacturing Technology, London, 2001, 18:586-590
[7] L. Li, J.Y.H. Fuh, Y.F. Zhang, A.Y.C. Nee: *Application of genetic algorithm to computer-aided process planning in distributed manufacturing environments*, Robotics and Computer-Integrated Manufacturing, 2005, 21:568-578

Izvornoznanstveni članak

Original scientific paper

Adresa autora / Authors' address:
Prof. dr. sc. Tonči Mikac, dipl. ing.
mr. sc. Dalibor Blažević
Domagoj Crljenko, dipl. ing.
Sveučilište u Rijeci, Tehnički fakultet
Vukovarska 58
HR-51000 Rijeka, Hrvatska