UDK 621.833.01

Optimiranje zobniških gonil
Optimisation of Gear Assemblies

JANEZ GAŠPERŠIČ - JOŽE FLAŠKER

V članku je opisan postopek optimiranja konstrukcijskih parametrov večstopenjskih zobniških gonil z uporabo genetskega algoritma. Predstavljeno je delovanje genetskih algoritmov in uveden posplošen model gonila. Glavna prednost postopka sta uporabnost in stabilnost pri široki paleti problemov.

The paper presents a method for optimisation of design parameters of multi-stage gear assemblies based on the genetic algorithm. The genetic algorithm is presented and a simplified gear assembly model is introduced. The main advantage of the presented optimisation procedure is its stability and applicability to many optimisation problems.

0 UVOD

Pri konstruiranju zobniških gonil moramo predvsem zagotoviti njihovo funkcionalnost v predpisani dobi trajanja. Poleg tega si prizadevamo za minimizacijo velikosti gonil in njihovih sestavnih delov, kar vodi k minimalni teži, majhni porabi materiala in nizki ceni. Pojavijo se številne geometrijske, kinematične, trdnostne, tribološke, tehnološke in ekonomske omejitve, ki jih moramo pri tem upoštevati. Posplošeno postopek sinteze obsega:

- definiranje funkcionalnosti, omejitev in izbiro kriterija optimizacije,
- ločitev parametrov na odvisne in neodvisne ter izgradnjo modela gonila na podlagi podanih zahtev,
- iterativen proces iskanja najugodnejše konstrukcijske variante s sprotno analizo modela.

V prvem koraku podamo konfiguracijo gonila, režime obratovanja, prenosne moči, predpisane vrtilne hitrosti, prestavna razmerja, medosja in drugo. Naloga optimizacijskega algoritma je poiskati tako kombinacijo modulov, števil zob, širin zobnikov in kotov poševnosti ozobja, da namenska funkcija zavzame najmanjšo vrednost.

0 INTRODUCTION

The primary goal when designing gear assemblies is to ensure their functionality for a certain service-life. In oder to reduce the production cost by rational use of the material it is often required that the overall dimensions of the assembly as well as the dimensions of its parts have to be optimised with respect to the minimum weight. One needs to take into account the geometric, kinematic, mechanical, tribological, technological and economic constraints. The generalised optimisation procedure consists of:

- the definition of functionality, constraints and selection of the optimisation criteria;
- identification of independent variables and definition of the gear assembly model;
- an iterative search process to find a theoretically ideal gear assembly design based on comprehensive model analysis.

In the first step, the assembly configuration, operating modes, transmitted powers, prescribed angular velocities, speed ratia, centre distances and other prescribed features of the gear assembly have to be specified. The purpose of the optimisation is to find a proper combination of gear moduli, number of teeth, teeth widths and tooth helix angles, at which the objective function reaches the minimum value.

1 POSEBNOSTI OPTIMIZACIJE ZOBNIŠKIH GONIL

V literaturi najdemo mnogo primerov optimizacij zobniških dvojic posameznih dvojic kakor tudi posameznih izvedb večstopenjskih zobniških gonil [6]. Največkrat je prikazana uporabnost determinističnih (običajno gradientnih) metod optimizacije. Uporablja se več optimizacijskih kriterijev, pogosto kriterij minimalne vsote prostornine delilnih valjev:

1 PARTICULARITIES OF THE GEAR ASSEMBLY OPTIMISATION

Several gear pair optimisation procedures, as well as optimisation procedures for selected types of multi-stage gear assemblies have been proposed in the past [6]. These are based on deterministic (usually gradient) optimisation methods. Several optimisation criteria can be used, most commonly the criterion of the minimum sum of pitch cylinder volumes:

$$F = \sum_{i=1}^{n} \left(m_{ni} z_i \cos(\beta_{oi}) \right)^2 b_i + \sum_{i=1}^{m} P_i \qquad (1).$$

Simbol P_i označuje funkcije omejitev, ki obsegajo množico dodatnih parametrov (koeficiente profilnih premikov, stopnje profilnega ubiranja, lastnosti materialov in drugo).

Posamezni parametri, ki bistveno vplivajo na konstrukcijo gonila (števila zob, moduli, medosja), lahko zavzamejo le celoštevilčne, standardizirane oziroma poprej izbrane vrednosti. Namenska funkcija postane zaradi tega nezvezna in dobi značilno stopničasto obliko.

Gradientne metode niso najprimernejše za reševanje takih problemov, saj nezmožnost natančnega določanja odvodov namenske funkcije bistveno omejuje njihovo uporabnost. Zato se postopek izvaja v več fazah, ki vedno vključujejo tudi naključno iskanje. V vsakem primeru je treba izračunati vrednost namenske funkcije v velikem številu točk. To kaže, da standardne metode na tem področju ne morejo izkoristiti vseh svojih prednosti, kar dopušča uporabo alternativnih metod optimiranja.

2 GENETSKI ALGORITEM OPTIMIZACIJE

Genetski algoritem je postopek optimizacije, ki temelji na posnemanju osnovnih načel naravnega razvoja. Glavne značilnosti genetskih algoritmov:

- postopek vodijo verjetnostna pravila in ne deterministična načela;
- uporabljajo karakteristično binarno kodiranje nabora parametrov in ne računajo neposredno s parametri;
- iskanje se ne prične v eni točki, ampak hkrati na celi množici točk;
- potrebujejo le funkcijsko vrednost namenske funkcije v posamezni točki, ne pa tudi vrednosti odvodov ali kakšnih drugih informacij.

Genetski algoritmi potrebujejo razmeroma veliko število izračunov funkcijske vrednosti namenske funkcije. Zato hitrost konvergence pri gladkih unimodalnih funkcijah, katerih lastnosti razmeroma dobro poznamo, ni velika v primerjavi z bolj determinističnimi metodami. Glavna prednost je v robustnosti. To pomeni, da nezveznost in večmodalnost namenske funkcije ne motita v tolikšni meri in je zato pričakovati konvergenco proti globalnemu ekstremu pri široki paleti funkcij [1].

Značilno za genetske algoritme je kodiranje parametrov. Najpreprostejše je binarno kodiranje, ki se običajno uporablja. Množico neodvisnih spremenljivk namenske funkcije kodiramo tako, da zapišemo vsako spremenljivko v binarni obliki, zapise pa nato nanizamo enega poleg drugega. Tako dobljen niz popolnoma določa neko konstrukcijsko varianto. Imenujemo ga kromosom. Parametri so izraženi v diskretni obliki, kar je nadvse dobrodošlo pri optimiranju gonil.

Symbol P_i represents the constraint functions that take into account a number of additional parameters (profile adjustment coefficients, material properties, etc.)

Many parameters, which considerably affect the design of the gear assembly (the number of teeth, moduli, centre distances), are only allowed to take integer values, standardised or some explicitly prescribed values. In this way the objective function becomes non-uniform and has a typical step shape.

Gradient methods are not best suited for solving such problems because the derivatives of the objective function cannot always be analytically determined. Therefore, the optimisation procedure needs to be carried out in several stages, that always include the random search patterns. In all cases, it is necessary to evaluate the objective function at a significant number of parameter points. This shows that the usual methods of optimisation cannot be fully implemented in this specific field. This implies that the use of some alternative methods is necessary

2 GENETIC OPTIMISATION ALGORITHM

The genetic optimisation algorithm is an optimisation procedure, which is based on the imitation of the principles of natural evolution. Important characteristics of the genetic algorithms are:

- the procedure is governed by probabilistic transition rules and not by deterministic principles;
- characteristic binary coding of the parameter set is used, and parameters are not directly involved in computations;
- the search process does not begin from a single point, but from a population of points;
- at every point only the value of the objective function is required, without the need to determine gradients or any other information.

Genetic optimisation algorithms in principle require many objective function evaluations. Therefore, their efficiency cannot be directly compared with deterministic methods, in which the objective functions are smooth, uni-modal and well defined. The main advantage of the genetic optimisation algorithms is their robustness. This means, that non-uniform and multi-modal objective functions make it more difficult to find the global extreme on a much smaller scale, if compared to the gradient methods [1].

Parameter coding is a characteristic feature of genetic algorithms. The simplest and most commonly used is binary coding. A set of independent parameters of the objective function is coded in a form of binary strings which are recorded sequentially. The resulting binary string is called the chromosome, which fully describes the related design. In this way, the discrete nature of variable description is introduced, which is very convenient for the optimisation of gear assemblies.

Genetski algoritem simulira razvoj populacije organizmov. Najprej naključno izberemo začetno množico točk, ki jo, glede na analogijo z biološkimi sistemi, imenujemo začetna populacija. Posamezne točke pomenijo člane populacije. Nato vsakemu članu priredimo stopnjo prilagojenosti, ki določa verjetnost njegovega preživetja. Če uporabimo običajno terminologijo: vsaki začetni točki priredimo vrednost namenske funkcije. Velikost populacije običajno predpišemo vnaprej in se med optimizacijo ne spreminja [2].

Genetski algoritem skuša izboljšati sedanjo populacijo. Za kriterij uspešnosti običajno vzamemo povprečno vrednost prilagojenosti njenih članov, pomemben podatek pa je tudi raztros od tega povprečja. Vseskozi med izvajanjem algoritma si prizadevamo izboljšati povprečje in hkrati vzdržujemo dovolj velik raztros. Postopek poteka tako, da s procesom selekcije izbiramo člane osnovne populacije in jih preoblikovane s tako imenovanimi genetskimi operatorji uvrščamo v naslednjo generacijo.

Način selekcije je izredno pomemben za pravilno delovanje algoritma. Osnovni teorem genetskih algoritmov zahteva, da je verjetnost izbire posameznega člana tem večja, čim večja je njegova prilagojenost. Dobre rezultate pri praktičnem delu daje metoda stohastičnega vzorčenja ostanka brez zamenjave.

Izkazalo se je, da ni dobro, če je verjetnost izbire nekega člana premosorazmerna z njegovo prilagojenostjo. To bi omogočilo prevelik vpliv najuspešnejših članov in povzročilo konvergenco k lokalnim rešitvam. Prehitro konvergenco preprečimo tako, da pred postopkom selekcije popravimo prilagojenost vseh članov in umetno zmanjšamo razliko med uspešnimi in manj uspešnimi člani. Znanih je več metod skaliranja, praktične izkušnje pa dajejo izrazito prednost metodi rangiranja [4].

Pri tej metodi člane sortiramo glede na njihovo vrednost prilagojenosti. Popravljeno prilagojenost člana predstavlja njegova razvrščenost. Hitrost konvergence lahko uravnavamo, če prilagojenost še dodatno popravimo z linearno funkcijo spremenljive strmine.

Glavni genetski operatorji so: reprodukcija, križanje in mutacija. Reprodukcija je uvajanje znanega Darwinovega načela naravne selekcije, ki pravi da preživijo le najmočnejši organizmi. Praktično jo izvedemo tako, da člane, izbrane s postopkom selekcije, prenesemo nespremenjene v naslednjo generacijo. Postopek zagotavlja, da imajo točke z večjimi vrednostmi namenske funkcije večjo možnost, da sodelujejo kot temelj pri iskanju v naslednji iteraciji. S tem je do neke meje zagotovljeno stabilno povečanje povprečne prilagojenosti populacije, ne pa tudi zvečanje prilagojenosti najuspešnejših članov. Če bi uporabili samo ta operator, bi po nekaj generacijah populacijo sestavljale samo kopije najuspešnejšega člana iz začetne populacije.

The genetic optimisation algorithm simulates the population development of living organisms. At the start the group of points-members is randomly selected, which - by analogy to biological systems - is called the primary population. Individual members are represented by a single point. The fitness value is then assigned to every member, which determines the ability of the member to survive. In the optimisation terminology it may be stated that every point is assigned an objective function value. The population size is usually prescribed at the beginning and does not change during the optimisation process [2].

The genetic algorithm tends to improve the current population. The criterion of improvement is usually the average of member fitness values; other important information is also the standard statistic deviation from the average value. During the optimisation the average population fitness should be improved, while at the same time sufficient deviation needs to be maintained. The evolution process consists of the members selection according to some selection mechanism and then their transformation by the genetic operators into the next population generation.

The selection mechanism is very important for appropriate functioning of the algorithm. The fundamental theorem of genetic algorithms implies that the probability of member selection is larger, if its fitness value is also large. The most popular and efficient method is that of stochastic remainder sampling without replacement.

It has been shown that the probability of member selection should not be directly proportional to the member's fitness. That would allow for exaggerated influence of just a few of the fittest members and would result in a premature convergence to local peaks. Such domination may be prevented by using the fitness scaling function to reduce artificially the fitness differences between members. Several scaling procedures may be used, however the practical experience shows that the ranking method is the best [4].

By using this method, the members are sorted according to their fitness. The corrected member fitness is now represented by its rank. The convergence speed may also be controlled by using the additional linear rank correction function with variable gradient.

The fundamental genetic operators are reproduction, crossover and mutation. Reproduction is essentially the implementation of the well known Darwinian principle of natural selection, which predicts the survival of only the fittest members. It is easily implemented by copying the selected members to the next generation without any modifications. This ensures continuous growth of the population average fitness, however no improvements are made to the fittest of individuals. If only this operator is used, it results after a few generations in an unbalanced population consisting only of copies of the best fitted individual from the primary population.

Pri križanju dveh organizmov prihaja do nastanka novih organizmov. Vsak izmed novo nastalih članov podeduje nekatere lastnosti enega ter preostale lastnosti drugega starša. To je gonilni proces optimizacije, ker pripravlja vedno nove konstrukcijske rešitve na podlagi uspešnih znanih rešitev.

Najboljše praktične rezultate pri optimiranju gonil daje zvezni operator križanja [3]. Najprej z uporabo selekcije izberemo dva starša P1 in P2 iz trenutne populacije. Z generatorjem naključnih števil izberemo masko M, tako da je verjetnost logičnega stanja 1 vsakega posameznega bita v maski 50%. Splošno lahko operator križanja zapišemo v obliki:

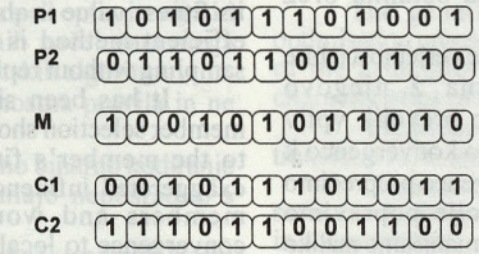
The crossover operator produces new members by mating two selected individuals. Every newborn member inherits some properties from both parents. This is the motive process of the genetic algorithm because it generates new, better solutions that are based on the best existing solutions.

The uniform crossover operator provides the best practical results [3]. Using this technique the two parents P1 and P2 are selected from the current population. Then the crossover bit mask M is randomly generated in such a way that the probability of each bit being set to 1 is equal to 50%. The crossover operator may be generally written in the form:

$$C1 = P1 \overset{e}{\vee} \left(\left(P1 \overset{e}{\vee} P2 \right) \wedge M \right)$$
$$C2 = P1 \overset{e}{\vee} \left(\left(P1 \overset{e}{\vee} P2 \right) \wedge \overline{M} \right) \qquad (2).$$

Pri križanju se ohranjajo tiste lastnosti (biti), ki so skupne obema staršema, vse preostale pa se naključno izmenjajo.

The crossover operator preserves the features (bits) that are common to both parents, while the remaining features are mixed randomly.



Sl.1. *Primer zveznega operatorja križanja*
Fig. 1. *An example of the uniform crossover*

Mutacija je proces, ki povzroča naključne napake pri razmnoževanju. To je zaželen pojav, kadar se pojavlja s pravilno dozirano verjetnostjo. Mutacija skrbi za stalno nastajanje alternativnih variant in s tem pomaga vzdrževati potreben raztros populacije.

Ko z uporabo selekcije in genetskih operatorjev določimo vso populacijo naslednje generacije, je prva iteracija postopka končana. Postopek ponavljamo, dokler ni izpolnjen kriterij za ustavitev postopka. Tega je razmeroma težko postaviti. Tu si ne moremo pomagati z naravnimi analogijami. Povprečna vrednost prilagojenosti članov se lahko med optimiranjem tudi zmanjša, nato pa spet močno zveča in je ni mogoče neposredno uporabiti kot končni kriterij. Običajno predpišemo največje število generacij, ki je glede na zmogljivost računalnika še časovno sprejemljivo, poleg tega pa nadziramo porast prilagojenosti najboljšega člana populacije. Če v predpisanem številu generacij ne pride do izboljšave, postopek ustavimo.

Mutation is an operator which introduces random errors into the member reproduction process. This is very useful, when the mutation occurs with appropriate probability. Mutation continuously generates alternate designs and helps to maintain sufficient population diversity.

The first iteration is finished after the next population is generated using the three genetic operators. The procedure is then repeated until the optimisation termination criterion is reached. It is difficult to find the appropriate termination criteria since there are no biological analogies. The average population fitness may decrease or strongly increase during the optimisation, and therefore cannot be directly used as a termination criterion. It is customary practice to define the number of generations with regard to the available computer capabilities and to track the fitness of the best population member. If there are no improvements in the prescribed number of generations, the algorithm can be stopped.

Poleg števila članov populacije sta osnovna kontrolna parametra genetskega algoritma verjetnost, s katero se pojavlja križanje, in verjetnost mutacije. Da dosežemo dobro učinkovitost postopka, je treba pazljivo nastaviti te parametre.

3 MODEL GONILA

Pred pričetkom optimizacije je treba definirati namensko funkcijo in omejitve. V ta namen v računalniškem pomnilniku zgradimo model večstopenjskega gonila, ki enolično opisuje njegove lastnosti. Z namenom, da izkoristimo prednosti, ki jih ponujajo genetski algoritmi posplošimo model do te mere, da dopustimo poljubno število zobnikov, gredi in režimov obratovanja.

Mogoče so seveda poljubne razmestitve zobnikov in različne povezave med njimi. Povezave se lahko spreminjajo glede na režime, v katerih obratuje gonilo. Predpisati je mogoče vse vrtilne hitrosti, prestavna razmerja, medosne razdalje, module, števila zob, širine zobnikov, kote poševnosti, materiale, ali pa izbiro nekaterih izmed njih prepustiti optimizacijskemu algoritmu. V tem primeru moramo seveda podati meje, v katerih dopuščamo spreminjanje. Vsakemu od omenjenih parametrov je mogoče prirediti tudi spisek dopustnih vrednosti, od optimizacijskega algoritma pa zahtevati, da izbira samo med njimi. Omejimo se na valjaste zobnike na negibnih vzporednih gredeh, model pa bo mogoče v prihodnosti posplošiti tudi na planetna gonila.

Uporaba objektno usmerjenih programskih jezikov omogoča preprosto izvedbo prilagodljivega modela. Posamezne elemente gonila definiramo kot objekte in jim priredimo pripadajoče lastnosti. Poglavitno vodilo pri snovanju modela gonila je bilo, da mora biti katerikoli podatek zapisan samo na enem mestu in to v sklopu tistega objekta, kateremu fizično ali logično pripada. Povezave med objekti omogočimo s kazalci. Takšna zasnova modela, skupaj z dinamičnim alociranjem pomnilnika omogoča, da modelu zlahka dodajamo ali odvzemamo nekatere elemente in ga tako interaktivno prilagodimo potrebam uporabnika.

Osnovni objekti so: *zobnik*, *os*, *režim*, *prenos*, *dvojica* in *material*.

Objekti tipa zobnik vsebujejo poleg osnovnih lastnosti (število zob, modul, koeficient premaknitve, kot poševnosti ozobja itn.) tudi kazalca na objekta tipa os in material. Objekt os določa premico v prostoru, na kateri je središče zobnika in vsebuje le koordinate te premice, medtem ko objekt material vsebuje številne podatke o lastnostih materiala, termični obdelavi in tehnološke podatke, ki so neposredno vezani na ta material. Objekt dvojica vsebuje prestavno razmerje, medosje in kazalca na soubirajoča zobnika. Pretok moči podajata objekta režim in prenos. Režim določa vstopno moč, število ur obratovanja v tem režimu in kazalec na objekt tipa prenos. Ta vsebuje kazalec na ustrezni zobnik in njegovo število vrtljajev v trenutnem režimu. Povezave med osnovnimi objekti modela gonila na sliki 2 so prikazane na sliki 3.

Apart from the number of population members, the fundamental control parameters of the genetic algorithm are the probability of crossover and mutation. In order to ensure the effectiveness of the optimisation it is necessary to choose carefully these parameters.

3 MODEL OF THE GEAR ASSEMBLY

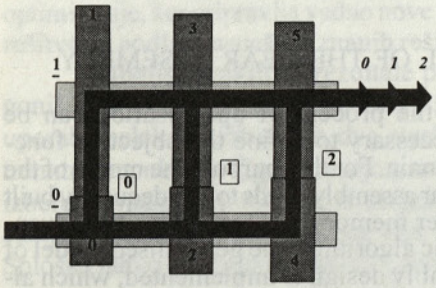
Before the process of optimisation can be started it is necessary to define the objective function and its domain. For that purpose the model of the multi-stage gear assembly needs to be adequately built in the computer memory. To emphasise the advantages of genetic algorithms the generalised model of the gear assembly design is implemented, which allows for the modelling of any number of gears, shafts and operating modes.

Any placement of gears and different links between them is allowed. Links may be defined separately for every operating mode. It is possible to explicitly define all angular velocities, speed ratia, centre distances, moduli, numbers of teeth, teeth widths, teeth helix angles, and materials, or to leave the selection of some parameters to the optimisation algorithm. In the latter case it is necessary to define the parameter limits. It is also possible to assign a list of allowed discrete values to some parameters, and the optimisation process should be able to choose only among them. Here, the discussion is limited only to spur and helix gears positioned on stiff parallel shafts. However, the model can also be extended to planetary gear assemblies.

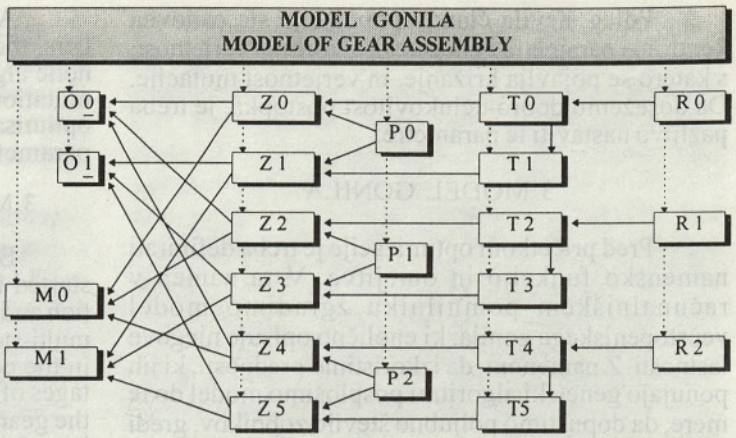
A flexible simulation model can be easily defined by using the object oriented programming. Individual elements of the design are defined as objects, with assigned apparent features. The individual design parameter is exclusively stored only once in the framework of the object where it physically and logically belongs. Links between objects are defined by pointers. Such model architecture, together with the dynamic computer memory allocation, enables easy customisation of the model by addition or removal of some model elements upon user request.

The basic objects of the gear assembly are: *gear*, *axis*, *operating mode*, *transmission*, *pair* and *material*.

Gear type objects comprise not only the typical gear parameters (number of teeth, module, coefficient of profile adjustment, teeth helix angle, etc.) but also pointers to the objects of type axis and material. The axis object determines the axis of the particular gear in space, and comprises only the coordinates of the centre - line, while the material object comprises numerous material properties of the gear, as well as the manufacturing data and thermal treatment data. The object pair includes speed ratio, centre distance and pointers to meshing gears. The power flow is defined by the object mode and transmission. The mode object contains the input power, anticipated number of hours for operating in that mode, and a pointer to the object of type transmission. The transmission object consists of the pointer to the assigned gear and its angular velocity in the current mode. Links between basic objects of the example design, shown in Figure 2, are illustrated in Figure 3.



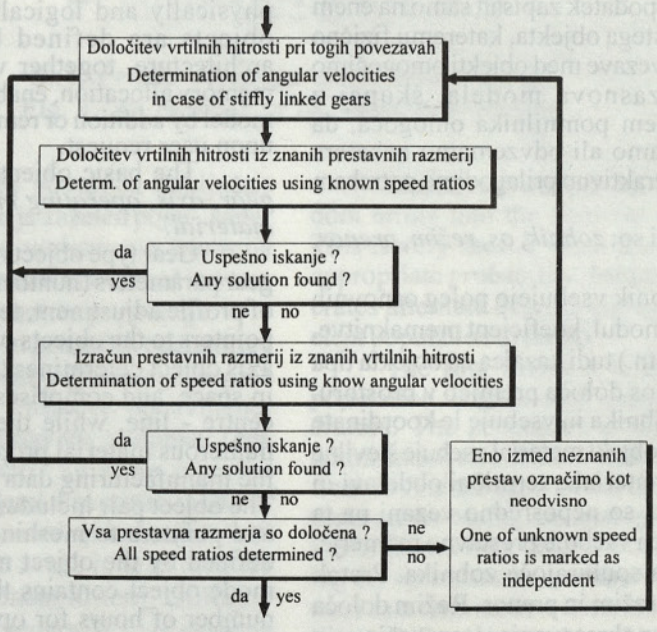
Sl. 2. *Model gonila*
Fig. 2: *Gear assembly model*



Sl. 3. *Povezave med osnovnimi objekti gonila*
(O - os, M - material, Z - zobnik, P - dvojica, T - prenos, R - režim)
Fig. 3. *Links between basic objects of the gear assembly*
(O - axis, M - material, Z - gear, P - pair, T - transmission, R - mode)

Očitno je, da vsi omenjeni parametri niso medsebojno neodvisni, zato je treba izdelati algoritem, ki loči odvisne spremenljivke od neodvisnih in prouči morebitne napake, nastale pri vnosu podatkov. To omogoča posredovanje izključno neodvisnih parametrov algoritmu optimizacije in interaktivno kontrolo vnosa podatkov. Vsakemu od osnovnih objektov dodamo funkcije, ki preverjajo, ali je mogoče z uporabo znanih parametrov tega objekta določiti kakšnega še ne določenega. Če je to mogoče, tega označijo kot znanega. Primer algoritma, ki usklajuje prestavna razmerja in vrtilne hitrosti, je prikazan na sliki 4.

It is clear that some of the design parameters depend on the other parameters. Therefore it is necessary to use an algorithm which separates the independent from the dependent parameters and also checks for possible input data inconsistencies. Such an algorithm enables direct input of independent parameters to the optimisation process and interactive control of all data input. Every basic object includes additional functions, to check whether the parameter belonging to the object has been already determined by some other object. If such a parameter is found, it is then marked as determined. The example of such an algorithm for determination of speed ratia and angle velocities when the number of teeth is not prescribed is shown in Figure 4.



Sl. 4. *Usklajevanje prestav in vrtilnih hitrosti*
Fig. 4. *Determination of speed ratia and angular velocities.*

V prvem koraku preverimo, ali poteka potek moči v kakšnem režimu prek dveh zobnikov, ki sta na isti osi in je znana vrtilna hitrost enega izmed njih. Če tak primer najdemo, označimo vrtilno hitrost drugega zobnika kot odvisen parameter. Nato preverimo, ali lahko določimo kakšno vrtilno hitrost iz znanega prestavnega razmerja in znane vrtilne hitrosti soubirnega zobnika. Če nam je do sedaj uspelo identificirati odvisnost kakšnega parametra, postopek ponovimo. V nasprotnem primeru poskušamo iz morebiti znanih vrtilnih hitrosti obeh zobnikov v kateremkoli od parov določiti njuno prestavno razmerje. Če tudi tu ni bilo uspeha, si izberemo eno od še nedoločenih prestav in jo označimo kot neodvisen parameter. Ko so znana (ali označena kot neodvisna) vsa prestavna razmerja je usklajevanje končano.

Postopki določanja neodvisnih parametrov in s tem tudi izračun odvisnih parametrov so zamudni in neprimerni za izvajanje pri vsakem izračunu namenske funkcije. Pospešimo jih tako, da ustvarimo sklad, na katerega beležimo izvedene operacije. Med naslednjimi iteracijami izvedemo le operacije, ki so na skladu in s tem prihranimo čas za iskanje.

In the first step, a check is made as to whether the power flows through two gears positioned on the same axis in any of the operating modes and whether the angular velocity of one gear is known. If this is so, then the two gears have equal angular velocities, and the other gear's angular velocity is marked as a dependent parameter. Then a check is made to decide whether it is possible to evaluate any angular velocity from the known speed ratio and from the known angular speed of the matching gear. If some dependent parameters have been identified, the process is repeated. If this is not the case, an attempt is made to determine speed ratia from the known angular velocities of matching gears. If the solution cannot be found, then one of the unknown speed ratia is explicitly defined and marked as an independent parameter. When all speed ratia are known (or marked as independent) the procedure is completed.

The algorithms for the determination of independent parameters and the evaluation of dependent parameters are time-consuming and thus unsuitable for execution at every evaluation of the objective function. To speed up the process, an evaluation stack is created, in which all operations executed are recorded. During the iteration process only the operations from the stack are executed, which reduces the search time.

4 POTEK OPTIMIZACIJE

Postopek optimizacije, ki se je najbolj izkazal pri eksperimentiranju z različnimi variantami, je prikazan na sliki 5.

V prvi fazi vse parametre obravnavamo kot realna števila. Praktično to pomeni, da vsak parameter kodiramo z 32 biti. Algoritem minimizacije geometrjiskih mer se nagiba k izbiri najkakovostnejših materialov, toplotnih obdelav in tehnoloških postopkov. V prvi fazi zato predpišemo najkakovostnejše materiale in postopke, pri čemer upoštevamo spiske dopustnih vrednosti.

Omejitve upoštevamo z omejitvenimi funkcijami. Te je treba izbrati zelo pazljivo. Prevelika strmina omejitvene funkcije povzroči, da je vrednost ustreznosti člana populacije, ki se naseli na prepovedanem območju, premajhna, da bi tak član preživel postopek selekcije. Naslednja generacija tako nima nobene informacije o obstoju prepovedanega območja in njen razvoj vodi le strmina na dovoljenem območju. Kadar je namenska funkcija dokaj monotona, dobimo zelo dobre rezultate z zrcaljenjem namenske funkcije prek mejne točke, kakor na primer pri omejitvi spodnje meje medosja:

4 THE OPTIMISATION PROCEDURE

Fig 5 illustrates the best optimisation algorithm, which has been determined by numerical experimentation with various algorithms.

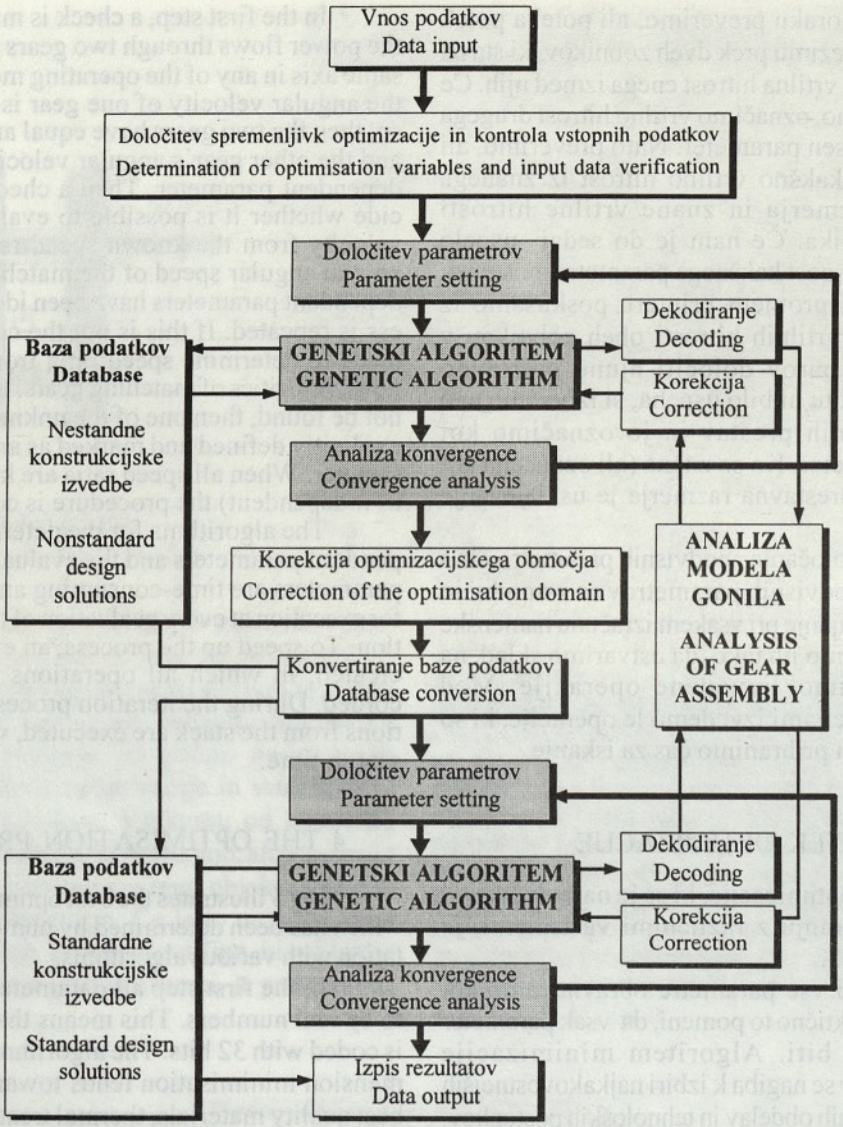
In the first step all parameters are considered to be real numbers. This means that each parameter is coded with 32 bits. The algorithm of geometric dimension minimisation tends towards the use of the best quality materials, thermal treatments and manufacturing procedures. Therefore in the first phase the best materials and procedures are prescribed, where the list of allowable values is taken into account.

The constraints are considered to be limiting functions, which need to be carefully chosen. If the rate of the limiting function is too high, the fitness value of any member inhabiting the forbidden design area too low to make its survival possible. The next generation therefore possesses no information on the forbidden area and the evolution depends only on the slope in the allowed design area. When the objective function is sufficiently monotonous, the best results are obtained when the objective function is mirrored over the limiting point, as is the case when limiting the lower value for centre distance:

$$P_i(a) = F(a_{\min} + (a_{\min} - a)) - F(a) = \\ = F(2a_{\min} - a) - F(a) \qquad (3).$$

Proces konvergence je v tem primeru hkrati voden z obeh smeri.

The convergence of the algorithm is here controlled from both directions.



Sl. 5. *Potek optimiranja gonila*
Fig. 5. *Flowchart of the gear optimisation procedure*

Daleč največji časovni obseg pripada algoritmu kontrole napetosti v korenu zoba in bočnega tlaka. Algoritem uporablja izračun po DIN 3990. Ti omejitvi upoštevamo tako, da pri izračunu namenske funkcije, ko je kršena ena izmed omejitev, računamo s potrebno širino zobnika.

Pri genetskem algoritmu uporabimo razmeroma majhno populacijo (20 do 50 članov) in postopek večkrat ponavljamo (3 do 10-krat). Najboljše rešitve posameznih iteracij hranimo v bazi podatkov. Začetno populacijo pred prvo iteracijo naključno izberemo. V nadaljnjih iteracijah sestavimo začetno populacijo iz baze, preostanek pa naključno izberemo.

Po vsaki iteraciji opravimo analizo konvergence genetskega algoritma in po potrebi opravimo kontrolne parametre. Če opazimo v zadnji fazi prevelik raztros, zmanjšamo verjetnost mutacije in križanja. S tem povečamo stopnjo neposredne reprodukcije. Prehitro

The most time-consuming is the determination of the maximum stress in the tooth root and the maximum Hertzian pressure between matching teeth. The algorithm presented uses German Industry Standard DIN 3990. The constraints of maximum allowed stress are regulated by the correction of the gear width.

By using the genetic algorithm, a relatively small population is involved (20 to 50 members) and the procedure is repeated few times (3 to 10 times). The best solutions of individual iterations are stored in the database. The primary population is randomly generated before the first iteration. In the following iterations the starting population is generated from the database and the rest is randomly generated.

After every iteration the convergence analysis is performed and the control parameters are adjusted, if necessary. If too large a diversity of population is observed in the last stage, the probability of mutation and crossover is lowered. Thus the direct

konvergenco in stagniranje postopka v zadnjih generacijah poskušamo uravnati s spremembo nagiba funkcije skaliranja.

Učinkovitost genetskega algoritma je v tej fazi pri tipskih gonilih manjša kakor pri standardnih metodah optimizacije. Prednosti se pokažejo pri večjem številu prostostnih stopenj in neobičajnih konfiguracijah, kjer je zanesljivost gradientnih postopkov majhna. Dodatne pospešitve lahko dosežemo z uporabo hibridnih tehnik, morda z uvedbo operatorjev lokalne optimizacije v genetski algoritem, kar pri obravnavanem algoritmu še ni raziskano.

V drugi fazi računamo z diskretnimi parametri. Optimizacijsko območje lahko pred tem zožimo, ni pa to nujno potrebno, saj dobi genetski algoritem informacije o položaju globalnega ekstrema od članov začetne populacije. To sestavimo pretežno iz najuspešnejših članov populacij v prvi fazi postopka. Zaradi spremembe kodiranja je pred tem treba konvertirati bazo podatkov. Genetski algoritem pokaže vse svoje prednosti v tej fazi postopka.

Izkaže se, da je območje, ki ga zavzema globalni ekstrem, dokaj široko. Zato ni pametno uporabiti enakega merila optimizacije v obeh fazah. V prvi fazi je primerno uporabiti kriterij minimalne vsote prostornin delilnih valjev ter s tem minimizirati maso gonila in porabo materiala. V drugi fazi pa izbiramo med kriterijem največje izkoriščenosti materialov in ekonomskimi kriteriji. Genetski algoritmi ponujajo tudi možnost večkriterijske optimizacije.

4 SKLEP

Genetski algoritmi so zanimiv nadomestek standardnim metodam optimizacije pri konstruiranju večstopenjskih gonil. Število potrebnih izračunov funkcijske vrednosti se pri tipskih problemih le zmerno poveča, pridobimo pa prožnost pri reševanju vrste problemov. Čas, potreben za optimizacijo gonila, znaša na osebnem računalniku (486DX50) 2 do 5 minut za vsako dvojico. Pričakujemo, da se bo z nadaljnjim povečevanjem zmogljivosti računalnikov uporabnost algoritma še povečala.

Zahvala

Delo je bilo opravljeno na Fakulteti za strojništvo Univerze v Mariboru, financirano s pomočjo Ministrstva za znanost in tehnologijo Republike Slovenije.

reproduction rate is increased. Premature convergence and stagnation of the procedure in the last generations is prevented by the change of the scaling function slope.

At this stage, the effectiveness of the genetic algorithm applied to typical gear assemblies is worse, if compared to the classical methods of optimisation. However, the genetic algorithms perform much better in the case of a larger number of independent parameters and unusual design configurations. Further improvements can be achieved by use of hybrid genetic operators, which need to be properly investigated.

In the next phase the discrete parameter values are used. It is not necessary to narrow the optimisation domain, since the genetic algorithm obtains information about the global extreme positions from the primary population members. The primary population consists of the best members from the first phase of the process. Due to the change of coding it is necessary to convert the database accordingly. All advantages of the genetic algorithm are demonstrated to the full extent in this phase of the process.

The solution domain around the global extreme is relatively wide, which necessitates the use of different optimisation criteria in the second phase. In the first phase it is advisable to use the criterion of the minimum sum of pitch cylinder volumes, which results in optimal assembly mass. In the second phase, criteria such as the rational use of material and the manufacturing cost consideration are more appropriate. The genetic algorithm also allows for multi-criteria optimisation.

4 CONCLUSION

The genetic algorithms present an interesting alternative to the usual methods of optimisation of multi-stage gear assemblies. The number of required evaluations of the objective function, in the case of typical gear assemblies only modestly increases, however the algorithm is more flexible and can be applied to a variety of optimisation problems. The computer execution time needed for optimisation of a gear assembly is 2 to 5 minutes for each gear pair (PC486DX50). It is expected that with further improvement of the computational capabilities the usefulness of the genetic algorithms will greatly increase.

Acknowledgements

The work was performed at the Faculty of Mechanical Engineering, University of Maribor, and financed by the Ministry of Science and Technology of Republic of Slovenia.

6 LITERATURA
6 REFERENCES

[1] Goldberg, D. E.: Genetic algorithms in search optimization & machine learning. Addison -Wesley Publishing Company, Inc., 1989.

[2] Goldberg, D. E.: Sizing populations for serial and parallel genetic algorithms. Proceedings of the Third International conference on genetic algorithms, Morgan Kaufmann Publishers, Inc., 1989.

[3] Syswerda, G.: Uniform crossover in genetic algorithms. Proceedings of the third International conference on genetic algorithms, Morgan Kaufmann Publishers, Inc., 1989.

[4] Whitley, D.: The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. Proceedings of the third international conference on genetic algorithms. Morgan Kaufmann Publishers, Inc. 1989.

[5] Arora, J. S.: Introduction to optimum design. McGraw-Hill Book Company, 1989.

[6] Prayoonrat, S.,Walton, D.: Practical approach to optimum gear train design. Computer aided design, 1988.

[7] Gašperšič, J.: Uporaba genetskih algoritmov pri optimiranju večstopenjskih zobniških gonil. Magistrsko delo, Univerza v Mariboru, Fakulteta za strojništvo, Maribor 1995.

Naslov avtorjev: mag. Janez Gašperšič, dipl. inž.
prof. dr. Jože Flašker, dipl. inž.
Fakulteta za strojništvo
Univerze v Mariboru
Smetanova 17
2000 Maribor

Authors' Addresses: Mag. Janez Gašperšič, Dipl. Ing.
Prof. Dr. Jože Flašker, Dipl. Ing.
Faculty of Mechanical Engineering
University of Maribor
Smetanova 17
2000 Maribor, Slovenia

Prejeto: 22.4.1996
Received:

Sprejeto: 30.10.1996
Acepted: