Multi-speed Gearbox Synthesis Using Global Search and Non-convex Optimization

Chiara Piacentini[2(✉)], Hyunmin Cheong[1(✉)], Mehran Ebrahimi[1], and Adrian Butscher[1]

[1] Autodesk Research, 661 University Avenue, Toronto, ON M5G 1M1, Canada
{hyunmin.cheong,mehran.ebrahimi,adrian.butscher}@autodesk.com
[2] Augmenta Inc., 106 Front St E, Toronto, ON M5A 1E1, Canada
chiara@augmenta.ai

Abstract. We consider the synthesis problem of a multi-speed gearbox, a mechanical system that receives an input speed and transmits it to an outlet through a series of connected gears, decreasing or increasing the speed according to predetermined transmission ratios. Here we formulate this as a bi-level optimization problem, where the inner problem involves non-convex optimization over continuous parameters of the components, and the outer task explores different configurations of the system. The outer problem is decomposed into sub-tasks and optimized by a variety of global search methods, namely simulated annealing, best-first search and estimation of distribution algorithm. Our experiments show that a three-stage decomposition coupled with a best-first search performs well on small-size problems, and it outmatches other techniques on larger problems when coupled with an estimation of distribution algorithm.

Keywords: Global search · Best-first search · Stochastic search · Evolutionary algorithms · Non-convex optimization.

1 Introduction

As demonstrated by the rich literature available [7,13,23,25,33,40], the ability to generate optimal designs is of crucial importance in various engineering applications, since it can lead to significant cost reduction or increased quality of the designed product. In this paper, we focus on the configuration optimization of a specific type of multi-component mechanical system, namely a multi-speed gearbox, which is particularly challenging due to its discrete and bi-level nature.

A gearbox is a mechanical system that transmits and converts an input speed to one or multiple output speeds by means of a series of rotating elements such as shafts and gears. Depending on the arrangements of the gears, we have two different models: the planetary gearbox, used to achieve automatic transmission, and the gear-pairs model, used in manual transmission. Here we only consider the manual transmission, often used in heavy-duty systems due to its higher

© Springer Nature Switzerland AG 2020
E. Hebrard and N. Musliu (Eds.): CPAIOR 2020, LNCS 12296, pp. 381–398, 2020.
https://doi.org/10.1007/978-3-030-58942-4_25

durability. A power source rotates the input shaft with an input velocity, which in turn rotates the next connected shaft, propagating the speed throughout the gearbox and towards the output. Pairs of shafts are connected through gear-pairs with different radii. The size of the gears' radii determines the speed change between the connected shafts. Typically, a gearbox is designed to convert the input speed into multiple output speeds according to several transmission ratios. This is achieved by having multiple shafts and multiple gear-pairs connecting those shafts. When a desired transmission ratio is selected, the associated gear-pairs are activated via clutches. Objectives considered for gearbox design can be the deviation of the transmission ratios from the nominal values, the power capacity, and the volume or the mass of the gearbox. The gearbox synthesis involves choosing the optimal configuration of shafts and gear-pairs, as well as their parameters such as gears' radii, with respect to the objectives.

The current work formalizes the gearbox synthesis as a bi-level optimization problem. The inner problem (or parameter optimization) takes a gearbox configuration as input and finds the position and size of the gears and shafts within given boundaries. This problem can be cast onto a non-convex continuous optimization problem and solved using state-of-the-art solvers. The outer optimization generates different configurations and we can solve it as a single step or as a decomposed set of sub-tasks. For both these approaches, we implement a variety of algorithms: simulated annealing (SA) and best-first search (BFS), which exploit a formulation of the configuration problem as a state transition system, and an estimation of distribution algorithm (EDA), from the evolutionary algorithms family. While in literature the gearbox synthesis is often modelled as a state transition system, in this paper we explore novel models and adapt them to better suit the search algorithms used. In summary, the contributions of this paper are:

1. formalization of multi-speed gearbox synthesis as a bi-level optimization problem;
2. presentation of two different decomposition approaches of the problem;
3. modelling of different sub-problems as state transition models;
4. application of BFS for multi-speed gearbox synthesis considering both lower and upper bounds;
5. development of an EDA for multi-speed gearbox synthesis.

The paper is organized as follows. Section 2 provides a literature review on gearbox synthesis, while Sect. 3 presents the notation and the problem definition. In Sect. 4 we describe two ways in which we can decompose the problem and we summarize each sub-problems. In Sect. 5 we formulate the sub-problems as state transmission models. Global search algorithms used in this work are described in Sect. 6. The experimental evaluation of our approach is shown in Sect. 7. Section 8 concludes our paper.

2 Literature Review

Gearbox design has been extensively studied in the mechanical engineering literature and different variations of the problem have been considered. A large body

of work deals with the parameter optimization problem, with the assumption that a fixed configuration is given as input. The problem can be formulated as a constraint satisfaction problem [26], and many works attempt to solve it using heuristic and meta-heuristic methods, such as local search [29,44], simulated annealing [17], genetic algorithms [4,11,30,43] and particle swarm optimization [31]. Multi-objective versions of the problem can also be found [10,15,27,41].

Gearbox configurations are often represented using graphs [42] and configuration synthesis is typically performed using *formal grammars*, i.e. a sets of rules that combine a finite set of elements to obtain a potentially infinite set of entities [9,24]. In the context of design synthesis, grammar rules offer a way to define how a design can be modified [6] and they have been widely used for gearbox generation [20,21,32,34,36]. Tsai et al. consider the configuration optimization of a planetary gearbox, where configurations are generated manually and duplicates are detected using graph isomorphism [36]. Schmidt & Chase developed a set of grammar rules that can be applied to generate several configurations of a planetary gearbox [32], while Li et al. propose a software that generates sketches of planetary gearboxes by modifying an initial design [20]. The set of rules developed by Lin et al. act on a graph representing the configuration of a manual transmission gearbox [21]. The graph is labelled to capture the relative position of gears and SA is used to generate candidate configurations and to fix some of the components' parameters. When a candidate is generated, collisions between elements are checked according to a set of constraints. The constraints identified by the authors, however, do not guarantee to avoid all possible collisions. Swantner & Campbell use an exhaustive tree search algorithm to generate a gearbox that can have a single transmission ratio, resulting in a fairly small system. The parameter optimization problem includes bending and stress constraints [34]. In the context of computational design synthesis, a complete search is used to study the quality of grammar rules [18]. Departing from this approach, Pomrehn & Papalambros consider the optimization of a gear train that outputs a single velocity as a mixed-integer non-linear programming model [28]. Berx et al. study a manual multi-speed gearbox and generate different configurations using constraint programming. A clustering procedure identifies promising candidate configurations, for which feasibility and objective value are calculated [2].

3 Problem Description

We consider the generation of manual multi-speed gearboxes and impose geometric constraints on gears and shafts, as proposed by Berx et al. [2].

3.1 Notation

Consider a gearbox consisting of a set of shafts $\mathbf{S}$ and a set of gear-pairs, i.e. gears connected with each other, $\mathbf{P}$. Each pair $p \in \mathbf{P}$ is comprised of an input gear g_p^i and an output gear g_p^o. We call $\mathbf{G}$ the union of all the input and output gears and we denote with s_g the shaft in which a gear g is situated, and s_p^i

and s_p^o the input and the output shafts of a gear-pair p, respectively. We call $\mathbf{C}$ the union of all the shafts and gears. We assume that all the shafts and gears are cylinders aligned along the z-axis. Each component $c \in \mathbf{C}$ is defined by the coordinates of the center of the bottom face of the cylinder (x_c, y_c, z_c), a radius ρ_c and a length (or thickness) l_c. We have set of transmission ratios Ω, where a transmission ratio ω is a real number indicating the desired ratio of the input and output speed.

A gearbox layout can be represented as an s-t multi-graph $\mathcal{G} = \langle \mathbf{S}, \mathbf{P} \rangle$ where the nodes $\mathbf{S}$ are the shafts and edges $\mathbf{P}$ are the gear-pairs. A source node s is identified as the input shaft, while a sink t is the output shaft of the system.

The graph is required to contain at least $k = |\Omega|$ simple paths from s to t. Each simple path π is a sequence of edges $\pi = (p_1, ..., p_n)$, with $p_i \in \mathbf{P}$ and all the vertices are distinct, producing a transmission ratio:

$$\omega_\pi = \prod_{j=1}^{n} \frac{\rho_{g_{p_j}^i}}{\rho_{g_{p_j}^o}} \tag{1}$$

where $\rho_{g_{p_j}^i}$ and $\rho_{g_{p_j}^o}$ are the radii of the input and output gears for gear-pair p_j.

Given the set of transmission ratios Ω, we call assignment Π^Ω the set of 2-tuples, containing a transmission ratio and a path $\Pi^\Omega = \{(\omega, \pi), \forall \omega \in \Omega\}$. We call configuration $\mathcal{C} = (\mathcal{G}, \Pi^\Omega)$ of a gearbox the 2-tuple consisting of the gearbox layout $\mathcal{G}$ and an assignment Π^Ω.

Figure 1 shows an example of a gearbox with an input shaft s^i, an output shaft s^o and an intermediate shaft s. Two gear-pairs are connected between s^i and s (p_1 and p_2) and two between s and s^o (p_3 and p_4). There are four possible paths between s^i and s^o: $\pi_1 = (p_1, p_3)$, $\pi_2 = (p_1, p_4)$, $\pi_3 = (p_2, p_3)$, and $\pi_4 = (p_2, p_4)$, corresponding to four different transmission ratios $\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4\}$. An example of assignment can be $\Pi^\Omega = \{(\omega_i, \pi_i), \forall i = 1, ..., 4\}$.

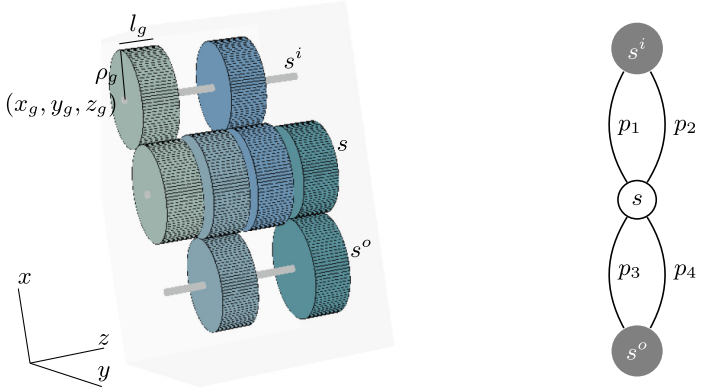


Fig. 1. Graph representation of the gearbox

3.2 Parameter Optimization

Let $\mathcal{C} = (\mathcal{G}, \Pi^{\Omega})$ be a given configuration, where shafts, gear-pairs and their input and the output are known. We want to find the components' parameters $(x_c, y_c, z_c, \rho_c, l_c)$ that minimize the total mass of the gears and produce the desired transmission ratios. If we assume that all the gears have a constant thickness (l_g) and are made of same material, the objective function is:

$$J_{\mathcal{C}}(\rho) = \sum_{g \in \mathbf{G}} \rho_g^2 \tag{2}$$

The system must satisfy several constraints based on design requirements and physical feasibility. For each pair in the assignment $(\omega, \pi) \in \Pi^{\Omega}$ the transmission ratio produced by $\pi = (p_1, ..., p_n)$ is within a given ϵ to the desired value ω:

$$\left| \prod_{j=1}^{n} \frac{\rho_{g^i_{p_j}}}{\rho_{g^o_{p_j}}} - \omega \right| \leq \epsilon \qquad \forall (\omega, \pi) \in \Pi^{\Omega}, \tag{3}$$

Every component $c \in \mathbf{C}$ must be contained inside a parallelepiped with maximum sizes $(x_{\max}, y_{\max}, z_{\max})$ and its radius has $\rho_{\min,c}, \rho_{\max,c}$ as limits:

$$\rho_c \leq x_c \leq x_{\max} - \rho_c \qquad \forall c \in \mathbf{C} \tag{4}$$
$$\rho_c \leq y_c \leq y_{\max} - \rho_c \qquad \forall c \in \mathbf{C} \tag{5}$$
$$0 \leq z_c \leq z_{\max} - l_c \qquad \forall c \in \mathbf{C} \tag{6}$$
$$\rho_{\min,c} \leq \rho_c \leq \rho_{\max,c} \qquad \forall c \in \mathbf{C} \tag{7}$$

A gear g must be placed on its connected shaft s_g:

$$x_g = x_{s_g} \qquad \forall g \in \mathbf{G} \tag{8}$$
$$y_g = y_{s_g} \qquad \forall g \in \mathbf{G} \tag{9}$$
$$z_g \leq z_{s_g} \qquad \forall g \in \mathbf{G} \tag{10}$$
$$z_g + l_g \leq z_{s_g} + l_{s_g} \qquad \forall g \in \mathbf{G} \tag{11}$$

In addition, we impose that the gears of each gear-pair must touch:

$$z_{g^i_p} = z_{g^o_p} \qquad \forall p \in \mathbf{P} \tag{12}$$
$$\rho_{g^i_p} + \rho_{g^o_p} = \sqrt{(x_{s^i_g} - x_{s^o_g})^2 + (y_{s^i_g} - y_{s^o_g})^2} \qquad \forall p \in \mathbf{P} \tag{13}$$

Finally, we require that two elements cannot occupy the same position in space. For two components $c, d \in \mathbf{C}$ that are not a gear and the shaft in which the gear is placed on, there exist two parameters $\lambda_{c,d}$ and $\alpha_{c,d}$, such that:

$$\lambda_{c,d}(z_c - z_d - l_d) + (\alpha_{c,d} - \lambda_{c,d})(z_d - z_c - l_c) + (1 - \alpha_{c,d})[(x_c - x_d)^2 + (y_c - y_d)^2 - (\rho_c + \rho_d)^2] > 0 \tag{14}$$
$$0 \leq \lambda_{c,d} \leq \alpha_{c,d} \leq 1 \tag{15}$$

This is the continuous formulation of the non-overlapping constraint obtained from the Lagrange duality applied to the distance determination problem [2,3].

Given a configuration $\mathcal{C}$, we define the parametric optimization problem as:

$$\min_{\rho} J_{\mathcal{C}}(\rho) \qquad (\mathcal{I}(\mathcal{C}))$$
$$s.t. \text{ Constraints (3)–(15)}$$

3.3 Configuration Optimization

When designing a gearbox configuration, we can impose a maximum limit on the number of shafts and the number of gears connecting any two shafts, $N_{\mathbf{S}}$ and $N_{\mathbf{P}}$, respectively. The configuration optimization problem is then:

$$\min_{\mathcal{C}} \mathcal{I}(\mathcal{C}) \qquad (\mathcal{O})$$

$$s.t. |\mathbf{S}| \leq N_{\mathbf{S}} \qquad (16)$$
$$\left|\mathbf{P}_{s_i,s_j}\right| \leq N_{\mathbf{P}} \qquad \forall s_i, s_j \in \mathbf{S} \qquad (17)$$

where $\mathbf{P}_{s_i,s_j}$ is the set of gear-pairs connecting shafts s_i and s_j. Similarly, we define $\mathbf{P}_s$ as the set of gear-pairs connected to s.

4 Problem Decomposition

This section shows two decomposition approaches. In the **two-stage decomposition approach**, the problem is divided into two sub-tasks:

A1. **Transmission ratio path assignment:** given a complete multi-graph $\mathcal{G}^{N_s,N_p} = (\mathbf{S}, \mathbf{P})$ with N_s shafts and N_p gear-pairs between every two shafts, and a set of transmission ratios Ω, generate an assignment Π^{Ω} using the s-t simple paths in $\mathcal{G}^{N_s,N_p}$. The layout $\mathcal{G}$ is the union of paths in the Π^{Ω};
A2. **Parameter optimization:** given the layout $\mathcal{G}$ and the assignment Π^{Ω}, solve $\mathcal{I}(\mathcal{G}, \Pi^{\Omega})$.

Since working with the complete multi-graph $\mathcal{G}^{N_{\mathbf{S}},N_{\mathbf{P}}}$ may result in a prohibitive large number of possible assignments, we consider an alternative **three-stage decomposition approach**, where we first select a sub-graph of $\mathcal{G}^{N_s,N_p}$ with at least k distinct s-t simple paths:

B1. **Graph generation:** generate a s-t multi-graph $\mathcal{G} = (\mathbf{S}, \mathbf{P})$ with at least k distinct simple paths from the input and output shafts;
B2. **Transmission ratio path assignment:** given a graph $\mathcal{G}$, find an assignment Π^{Ω} using the s-t simple paths in $\mathcal{G}$;
B3. **Parameter optimization:** given a graph $\mathcal{G}$ and an assignment Π^{Ω}, solve problem $\mathcal{I}(\mathcal{G}, \Pi^{\Omega})$.

4.1 Graph Generation

Graph generation entails finding a sub-graph $\mathcal{G}$ of $\tilde{\mathcal{G}}$ with at least k s-t simple paths. Given an edge-weighted graph $\tilde{\mathcal{G}} = (V, E)$, a source node $s \in V$, a sink node $t \in V$ and a positive number k, we call *min-k-simple paths* (mkSP) the problem of finding the minimum weighted edges $E' \subseteq E$, such that there exists k distinct simple paths π_i from s to t, whose edges belong to E': $\pi_i \subseteq E'$ for $i = 1, \ldots, k$. In the gearbox synthesis, the input is a complete multi-graph with $\mathcal{G}^{N_\mathbf{S}, N_\mathbf{P}}$ and the solution can provide a lower bound on the number of gears necessary to add to the system.

The problem can be seen as a particular type of coverage problem, namely *min-k-union* (mkU) [38]: given a set $\mathcal{U}$, a collection $\mathcal{S}$ of subsets of $\mathcal{U}$, and an integer number k, select k subsets of $\mathcal{S}$ to minimize the number of covered elements in $\mathcal{U}$. Min-k-union is the minimization version of the classical maximum coverage problem. While for maximum coverage, a greedy solution has a $(1-1/e)$ approximation guarantee, mkU is harder to approximate [8].

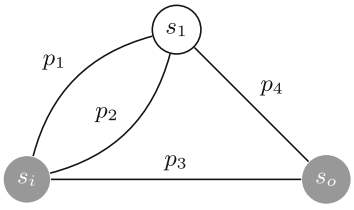
The reduction of mkSP to mkU is trivial: the set of all the edges E in mkSP is $\mathcal{U}$ in mkU and the set of all paths corresponds to $\mathcal{S}$. This, however, requires the identification of all the s-t simple paths in $\mathcal{G}$, which is a #P-COMPLETE task in the general case [37].

When generating graphs, we can leverage *graph isomorphism* to avoid the repeated evaluation of equivalent graphs. While it is not known if the detection of isomorphic graphs can be solved in polynomial time or if it is a NP-COMPLETE problem, efficient solvers are available [22].

4.2 Transmission Ratio Path Assignment

Given a graph $\mathcal{G}$ with n s-t simple paths and a set of transmission ratios Ω, the task requires to find an assignment Π^ω. If we have k transmission ratios, with $k \leq n$, we have $\frac{n!}{(n-k)!}$ assignments. To reduce the number of assignments that we need to check, we can identify those that lead to the same parameter optimization problem.

Consider the example in Fig. 2 with three transmission ratios $\omega_1, \omega_2, \omega_3$. The configuration contains three simple paths: $\pi_1 = (p_1, p_4), \pi_2 = (p_2, p_4), \pi_1 = (p_3)$ and six assignments can be made: $\{(\omega_1, \pi_1), (\omega_2, \pi_2), (\omega_3, \pi_3)\}$, $\{(\omega_1, \pi_1),$



	ω	π	edge-seq	labeling
Π^Ω	ω_1	π_1	(p_1, p_4)	(l_1, l_2)
	ω_2	π_2	(p_2, p_4)	(l_3, l_2)
	ω_3	π_3	(p_3)	(l_4)
Π'^Ω	ω_1	π_2	(p_2, p_4)	(l_1, l_2)
	ω_2	π_1	(p_1, p_4)	(l_3, l_2)
	ω_3	π_3	(p_3)	(l_4)

Fig. 2. Example of transmission ratio path assignments and their labeling

(ω_2, π_3), $(\omega_3, \pi_2)\}$, $\{(\omega_1, \pi_2)$, (ω_2, π_1), $(\omega_3, \pi_3)\}$, $\{(\omega_1, \pi_2)$, (ω_2, π_3), $(\omega_3, \pi_1)\}$, $\{(\omega_1, \pi_3)$, (ω_2, π_2), $(\omega_3, \pi_1)\}$, and $\{(\omega_1, \pi_3)$, (ω_2, π_1), $(\omega_3, \pi_2)\}$. If we have only a single transmission ratio, paths with the same lengths will lead to the same parameters. In our particular example, this allows us to check only three assignments: $\{(\omega_1, \pi_1)$, (ω_2, π_2), $(\omega_3, \pi_3)\}$, $\{(\omega_1, \pi_1)$, (ω_2, π_3), $(\omega_3, \pi_2)\}$, and $\{(\omega_1, \pi_3)$, (ω_2, π_2), $(\omega_3, \pi_1)\}$.

Formally, given a gearbox problem, a layout $\mathcal{G}$ and two assignments Π^{Ω} and Π'^{Ω} over a set of s-t simple paths in $\mathcal{G}$, we say that the two assignments are equivalent $\Pi^{\Omega} \equiv \Pi'^{\Omega}$ if problems $\mathcal{I}((\mathcal{G}, \Pi^{\Omega}))$ and $\mathcal{I}((\mathcal{G}, \Pi'^{\Omega}))$ produce the same optimal solution. To detect equivalent assignments, we fix an arbitrary order of the transmission ratios and label the edges of the associated paths in their order of appearance. Thus, two assignments with the same labeling are equivalent.

4.3 Parameter Optimization

The parameter optimization $\mathcal{I}(\mathcal{C})$ is a non-convex optimization problem. We solve it using the solver Ipopt [39]. Ipopt is a software specialized in large scale continuous non-linear optimization problems, based on primal-dual interior point method. For non-convex problems, it does not guarantee that the solution found is globally optimal nor that the problem is globally infeasible. As shown in previous work [2], Constraints (14)–(15) are hard to enforce and can cause the solver to get stuck in a region of local infeasibility. To alleviate this problem, we first consider a relaxation of the problem containing only Constraint (3). If the relaxation is infeasible, we also consider the original problem to be infeasible. Otherwise, we use its solution to initialize the values of the radii of the gears. If the solver finds that $\mathcal{I}(\mathcal{C})$ is infeasible, we restart the solver with a different random initialization of the parameters, up to a fixed maximum number of times, or until a feasible solution is found.

5 State Transmission Models

In this section, we present different transition systems that can be used to solve the sub-tasks of our problem. Similar to grammar rules [18,21], a transition system can be used to describe how an algorithm moves from one partial solution (a state) to another, using applicable actions.

More formally, a state transition system is defined by a set of states $\mathcal{S}$, a set of possible actions $\mathcal{A}$ and a transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$. An action can be applied to a state if the state satisfies some *preconditions*. Each transition is characterized by a cost function $\mathcal{Q}_{\mathcal{T}} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$. Given an initial state $\sigma_i \in \mathcal{S}$ and a set of goal states $\mathcal{S}_g \subseteq \mathcal{S}$, we want to find a sequence of applicable actions $\pi = (\alpha_1, ..., \alpha_n)$, such that after applying the actions from σ_i, we obtain a goal state $\sigma_g = \mathcal{T}(\mathcal{T}(\mathcal{T}(\sigma_i, \alpha_1), ...), \alpha_n) \in \mathcal{S}_g$, using a sequence of actions with minimal cost. In all the sub-tasks considered, the cost does not depend on the transition, but only on the state.

5.1 Graph Generation

A state is an s-t multi-graph $\mathcal{G} = (\mathbf{S}, \mathbf{P})$ with nodes $\mathbf{S}$ and edges $\mathbf{P}$. The source of the graph is the input shaft $s^i \in \mathbf{S}$ while the sink is the output shaft $s^o \in \mathbf{S}$. A goal state is defined as a graph containing $k = |\Omega|$ simple paths from s_i to s_o. We assign cost 0 to non-goal states, while the cost of a goal state is determined by the solution of the nested problem. We report here the primitive actions for this task. However, depending on the algorithm, we use a subset of such actions, or we define new actions as a sequence of these primitive actions. In the following, every action is identified by a name, where the subscript indicates the involved objects (gears, shafts, paths), its preconditions Φ, and the transition function $\mathcal{T}$ from a state $\mathcal{G} = (\mathbf{S}, \mathbf{P})$:

- a_{p,s_j,s_k}: add a gear-pair p between shafts $s_j, s_k \in \mathbf{S}$. $\Phi : |\mathbf{P}_{s_j,s_k}| < N_p$, $\mathcal{T} : (\mathbf{S}, \mathbf{P} \cup \{p\})$
- a_s: add a new shaft s. $\Phi : |\mathbf{S}| < N_s$. $\mathcal{T} : (\mathbf{S} \cup \{s\}, \mathbf{P})$
- d_p: delete edge $p \in \mathbf{P}$. $\Phi : p \in \mathbf{P}$. $\mathcal{T} : (\mathbf{S}, \mathbf{P}/\{p\})$
- d_s: delete shaft $s \in \mathbf{S}$. $\Phi : s \in \mathbf{S}$, $s \neq s^i$, $s \neq s^o$, $\mathcal{T} : (\mathbf{S}/\{s\}, \mathbf{P}/\mathbf{P}_s)$

5.2 Transmission Ratio Path Assignment

We consider two distinct models for the transmission ratio path assignment task. The first model, called **paths generation model**, incrementally builds paths adding edges, while the second, **assignment model**, finds all the s-t simple paths in a graph and tries to assign each transmission ratio to one of them.

Paths Generation Model. In path generation, a state is defined as a tuple $(\pi_1, ..., \pi_k)$, where $\pi_i = (p_1, ..., p_{n_i})$ is the sequence of gear-pairs associated with the transmission ratio ω_i, starting with shaft s^i. The sequence requires the gear-pairs to be connected: i.e. $s^i_{p_j} = s^o_{p_{j-1}}$, $\forall j = 2, ..., n_i$. A sequence may be empty. We say that a sequence is a *complete path* if it terminates with shaft s^o. A goal state is a state where all the sequences of gear-pairs are complete, distinct paths. A goal state uniquely identifies a transmission ratio path assignment. For this task, we only add or delete gear-pairs from a state $(\pi_1, ..., \pi_k)$ (shafts are automatically added if they are the input or the output of the gear-pair added):

- a_{p,π_i}: add gear-pair p at the end of π_i. $\Phi : s^i_p = s^o_{p_{n_i}}$, $|\mathbf{P}_{s^i_p,s^o_p}| < N_p$, $|\mathbf{S}| < N_s$, $\pi_i \cup p$ is a simple path. $\mathcal{T} : (\pi_1, ..., \pi_i \cup p, ..., \pi_k)$,
- d_{π_i}: delete the last element of π_i. $\Phi : \pi_i$ is not empty. $\mathcal{T} : (\pi_1, ..., \pi_i/p_{n_i}, ..., \pi_k)$

The cost of a non-goal state is determined by the solution of the relaxed version of the parameter optimization problem, which considers only the transmission ratios associated with complete paths. The cost of goal states is the cost of the full parameter optimization problem.

Assignment Model. For this model, given a graph $\mathcal{G}$ containing n simple paths from s^i to s^o, a state is a partial assignment $\tilde{\Pi}^{\Omega}$, i.e. a subset of Π^{Ω}. Actions assign a simple path to a transmission ratio, i.e. add an element (ω, π) to the partial assignment. A goal state is defined as a state where all the transmission ratios are assigned. The cost of a non-goal state is determined by the solution of the relaxed version of the parameter optimization problem over the partial assignment, while for goal states, the cost is the objective value of the full parameter optimization problem.

6 Global Search

This section describes three search algorithms used to solve the gearbox synthesis problem. For each search, both the two-stage and three-stage decomposition approaches are considered (see Fig. 3). Each sub-task is solved using transition systems described in the previous section and its solution is the input of the next sub-task. The first algorithm presented, our baseline, is SA [16]. Next, we consider a BFS [12] that keeps track of both the upper bound and lower bound of solutions, similarly to branch-and-bound [19]. Finally, an EDA, a population-based meta-heuristics approach is used.

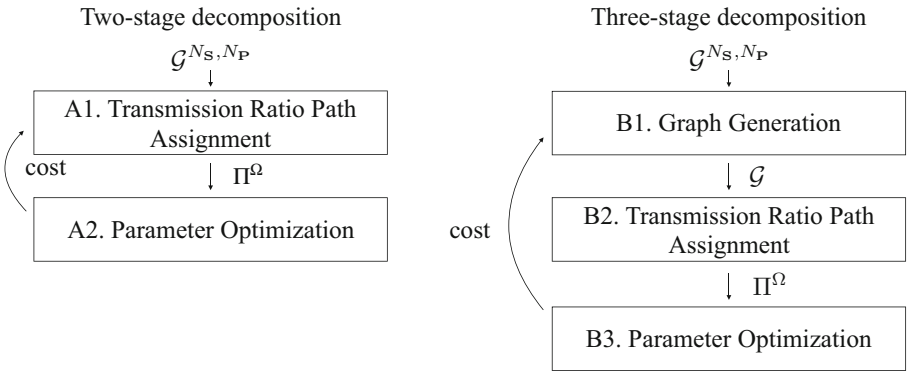


Fig. 3. Flowchart of the decomposition approaches

6.1 Simulated Annealing

In SA, states are feasible solutions and their cost corresponds to the cost of a solution. A new state is generated by randomly selecting one transition of the transition model and solving the subsequent sub-tasks. If this results in a non-feasible solution, the state is automatically rejected.

Two-Stage Decomposition. The search is initialized with randomly generated paths and the neighborhood is defined by the paths generation model in Sect. 5.2. We consider the following actions, obtained by concatenating the actions in Sect. 5.2:

- delete gear-pair p at position i in π: $d_{p,i,\pi} = d_{p_n,\pi}, d_{p_{n-1},\pi}, ..., d_{p_{i-1},\pi}, a_{(s^i_{p_{i-1}},s^i_{p_{i+1}}),\pi}, a_{p_{i+2},\pi}, ..., a_{p_n,\pi}$
- add gear-pair p at position i in π: $a_{p,i,\pi} = d_{p_n,\pi}, d_{p_{n-1},\pi}, ..., d_{p_{i-1},\pi}, a_{(s^i_{p_{i-1}},s^i_p),\pi}, a_{p,\pi}, a_{(s^o_p,s^o_{p_i}),\pi}, a_{p_i,\pi}, ..., a_{p_n,\pi}$
- replace gear-pair p_j with p_k in π: $r_{p_j,p_k,i,\pi} = d_{p_j,i,\pi}, a_{p_k,i,\pi}$

Three-Stage Decomposition. For this approach, we first generate a graph using the graph generation model in Sect. 5.1 with the actions devised by Konigseder et al. [18], which are a combination of actions in Sect. 5.1:

- create a new gear-pair between two existing shafts: a_{p,s_j,s_k}
- delete an existing gear-pair p: d_p
- create a new shaft and connect to two existing ones: $a_s, a_{p_i,s^i_{p_j},s}, a_{p_j,s,s^o_{p_j}}$
- delete a shaft: d_s
- replace a gear-pair: $d_p, a_s, a_{p_j,s^i_p,s}, a_{p_k,s,s^o_p}$

Starting from a randomly generated graph, the SA randomly selects one of the actions and finds a new graph. The transmission ratio-path assignment problem is solved heuristically by selecting a limited number of assignments, which are then used to solve the parameter optimization problem. The cost of the graph is the best objective value found among the sub-problems considered.

6.2 Best-First Search

In BFS, starting from the initial state, the algorithm selects a node for expansion based on an evaluation function f, which represents an estimation of the cost of the best solution. During *expansion* all the possible successor states are generated, *evaluated* and inserted into a priority queue. The process is repeated, selecting each time the state in the queue with the lowest f, until a goal state is retrieved from the queue. We consider here a tweaked version of BFS that also keeps track of an upper bound [5], which is simply the cost of the best incumbent solution found during the evaluation stage of the algorithm. This allows us to return the such feasible solution when we limit the running time of the algorithm. Our evaluation function f is defined as a lower bound of the cheapest configuration that can be achieved from a state σ and it is set to 0 at initialization. The optimality of the algorithm cannot be guaranteed because our parameter optimization problem is not solved to (global) optimality.

Two-Stage Decomposition. Similarly to SA, we solve the transmission ratio path assignment problem using the path generation model in Sect. 5.2. We start a search with a state with empty paths and incrementally add all possible edges to create the first path. When the path is complete, we solve the inner optimization problem to determine the cost of the state and we can start adding edges to create the next path. Notice that the only action that we need is the addition of a gear-pair. The evaluation function is calculated by running the relaxed solver on paths that are completed. In addition, we add a lower bound related to the number of gear-pairs that we need to add to the graph to have k paths. This is calculated by building the graph representing the gearbox layout as the union of the paths and estimating the number of edges necessary to have k s-t simple paths: if the graph has k s-t simple paths, this is the number of edges in the graph, otherwise, we take the number of edges plus one. We calculate the lower bound on the cost by multiplying such number by two (every edge is a pair of gears) and the square of the minimum radius of the gears.

Three-Stage Decomposition. We set the initial state to be a graph containing only the input s^i and the output s^o shafts and consider the actions in Sect. 5.1:

- create a new gear-pair between two existing shafts: a_{p,s_j,s_k}
- create a new gear-pair p between an existing and a new shafts s: $a_s, a_{p,s_j,s}$

Since we insert all the states explored in the queue and we start with an empty graph, we do not need actions that delete gear-pairs or shafts. We use graph isomorphism to detect duplicated states. When a goal state (a graph with k simple paths from s^i to s^o) is generated, we calculate the minimal cost of the graph. This cost is calculated by solving the transmission ratio path assignment problem. The evaluation function is calculated by estimating the minimum number of edges that we need to have a graph with at least k simple paths, similarly to the 2-stage decomposition approach. Since the cost of a graph is defined by $\mathcal{I}$, which is a non-convex problem, adding a gear-pair does not necessarily increase the cost of a configuration. For this reason, the evaluation function of a goal state is not the cost of the configuration but is the lower bound defined above. Our algorithm does not terminate when a goal state is found, but when the gap between lower and upper bound is 0 or all the states have been explored.

To solve the transmission ratio path assignment problem, we run another BFS using the assignment model in Sect. 5.2, where we start from a state where none of the transmission ratios is assigned and actions correspond to the assignment of a simple path to a transmission ratio.

6.3 Estimation of Distribution Algorithm

The last type of global search method is EDA, a class of meta-heuristic approaches based on the evolution of populations [14]. While typical evolutionary algorithms, such as genetic algorithms, use variation operators such as cross-over and mutation, EDAs use an explicit probability model to generate new solutions,

showing advantages in terms of performance, theoretical convergence, and capturing the structure of the problem space [14]. The probabilities are computed by directly using the frequency statistics of the selected top individuals from a population and indicate the likelihood of a particular solution being included in a set of top quality solutions based on prior observations. EDA has also been successfully applied to the configuration design of vehicle suspension systems [7]. As with the other algorithms, both two and three-stage decomposition approaches are implemented and EDA is used to solve the top-level task. The overall framework for the both approaches is the same except for the details in each of the steps, as detailed below.

i) Generation of Initial Population: As a first step, an initial population of individuals $\mathcal{P}$ is randomly generated.

In the two-stage decomposition approach, an individual is a tuple of k simple paths, each corresponding to a transmission ratio path assignment. Each path is constructed by randomly choosing an edge at a given node from a uniform distribution defined over all possible outgoing edges of the node, starting from the input shaft node s^i and ending at the output shaft node s^o.

In the three-stage decomposition approach, an individual is a graph $\mathcal{G}$. An individual is generated by randomly selecting an edge from a uniform distribution defined over all possible edges, and incrementally adding the selected edge to a null graph until the number of simple paths n is such that $k \leq n \leq n_{max}$. We impose an upper bound on the number of paths to limit the number of transmission ratio path assignments. If the last edge added results in $n > n_{max}$, we backtrack one step and add another edge until $k \leq n \leq n_{max}$.

ii) Evaluation and Selection: Each individual in the population is evaluated by solving the parameter optimization problem. For the three-stage decomposition approach, several transmission ratio path assignments are created exhaustively before parameter optimization. A subset of the population, $\mathcal{P}' \subseteq \mathcal{P}$, representing top t individuals, is selected. $|\mathcal{P}'|/|\mathcal{P}|$ is called truncation rate.

iii) Estimation of Probability Distribution: From $\mathcal{P}'$, the probability distributions of the edges in the individuals are estimated.

The probability model used for the two-stage decomposition approach is:

$$\mathbb{P}_{path,\pi_j}(\mathbf{P}) = \prod_{p \in \mathbf{P}} \mathbb{P}_{path,\pi_j}(p|s_p^i) \qquad \forall j = 1, ..., k \tag{18}$$

Here, a conditional probability distribution is assumed and the probability of a gear-pair p is dependent on its input shaft, s_p^i.

The probability model used for the three-stage decomposition approach is:

$$\mathbb{P}_{graph}(\mathbf{P}) = \prod_{p \in \mathbf{P}} \mathbb{P}_{graph}(p) \tag{19}$$

In other words, a univariate probability distribution is assumed and the probability of an edge p is determined independently from other edges.

iv) New Population Generation: A new population of individuals is generated using the same techniques as in Step i) with probability distributions estimated in Step iii). That is, instead of randomly selecting edges from uniform distributions to construct a graph or a path, the edges are sampled from the probability distributions in Eq. (19) or Eq. (18), depending on the approach.

v) Iterate Steps ii)–iv): Steps ii) to iv) are repeated with the newly generated population until an allocated number of iterations is reached.

7 Experimental Evaluation

Experiments are run on a single desktop computer with two Intel Xeon CPUs E5-2650 v2 2.60 GHz and 32G of RAM.

7.1 Experimental Setup

Datasets. We generate a first dataset with 45 synthetic problem instances (*synthetic dataset*). Instances have 4, 7 and 10 transmission ratios. We fix the minimum transmission ratio to 1, and the maximum to 2, 6 and 10, respectively. The values of the transmission ratios are generated randomly between the minimum and the maximum. The second dataset (*tremec dataset*) contains nine realistic problems, five with five transmission ratios and four with ten transmission ratios. The transmission ratios and the size of the gearbox are taken from the specifications found on the TREMEC website [35].

Implementation Details. The inner problem $\mathcal{I}(\mathcal{C})$ is solved with Ipopt [39], using the modeling language CasADi [1]. Isomorphic graphs are detected using the nauty library [22]. For EDAs, evaluations are run in parallel across 32 CPU threads available in the computer used. For the two-stage decomposition approach, we use population sizes of 192 for the five-ratio problems and 384 for the ten-ratio problems in the *tremec* dataset. For the three-stage decomposition approach, we use population size of 64, while using different time limits for evaluating each individual: three minutes for the five-ratio problems and six minutes for the ten-ratio problems. For all EDAs, the truncation rate and the number of iterations are set to 0.2 and 10, respectively. Both SA and BFS are run on a single thread. The temperature parameter in SA is set to 2000 and decremented every 1000 evaluations with a step size of 5. Time limits vary depending on the size of the problem and are reported in the next section. They are assumed to be tolerable waiting times from a design process perspective.

Table 1. Results on the *synthetic dataset*. We report the average and the standard deviation of objective value for problems that are solved by all algorithms (excluding those that cannot solve any problem). We report also the number of problems solved by each algorithm.

k		4						7						10					
max ω		2		6		10		2		6		10		2		6		10	
		μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
SA2	obj	1349	92	2868	336	3377	105	6821	581	13389	2083	22805	1449	-	-	-	-	-	-
	#	5	-	5	-	5	-	5	-	5	-	5	-	0	-	0	-	0	-
BFS2	obj	**1168**	99	**2015**	97	**2596**	121	1715	45	-	-	-	-	-	-	-	-	-	-
	#	5	-	5	-	5	-	5	-	0	-	0	-	0	-	0	-	0	-
SA3	obj	**1168**	100	2033	120	2644	145	1764	116	**3456**	268	**4427**	285	3161	608	**5337**	-	**6787**	-
	#	5	-	5	-	5	-	5	-	5	-	5	-	5	-	4	-	5	-
BFS3	obj	**1168**	99	**2015**	97	2597	121	**1640**	22	3858	307	6018	499	**2258**	364	7565	-	24206	-
	#	5	-	5	-	5	-	5	-	5	-	5	-	5	-	1	-	1	-

Table 2. Results on the *tremec dataset*. We report the average and the standard deviation of the objective value of each problem. The value marked with * is the objective value of the only solution found in all runs of the algorithm.

	5-transmission ratios										10-transmission ratios							
problem id	es42-5a		es52-5a		es60-5a		es60-5c		tr-3550		tr-t-10d		tr-t-10v		tr-to-10s		tr-to-10v	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
EDA2	23281	1307	23733	957	24294	1522	24468	1538	12670	761	-	-		-	-	-	-	-
SA2	28523	2102	28114	2363	27468	4187	28863	3966	14501	1797	-	-	-	-	-	-	466813*	0
BFS2	-	-	-	-	23797	265	24651	696	**10952**	200	-	-	-	-	-	-	-	-
EDA3	**21644**	473	**22014**	414	**21199**	0	**22674**	155	11067	69	**22891**	1247	**25936**	1343	**21844**	711	**22698**	631
SA3	25866	1957	26401	1773	25401	1993	24232	2618	11976	800	44105	10130	46084	20198	33093	9432	40640	10993
BFS3	31022	573	31370	552	30799	438	30792	538	10995	162	55458	1049	69725	1571	47989	1291	51338	1127

7.2 Results

We use the *synthetic dataset* to test the behaviour of the algorithms running on a single thread: BFS and SA. In Table 1 we report the average solution qualities and the number of problems solved for different groups of problems, using a time-limit of 1 h. The name of the algorithm is followed by 2 or 3, indicating the two-stage or three-stage decomposition approach, respectively. Every algorithm is run 3 times to account for the randomization in the algorithms. Both SA and BFS perform better when using the three-stage decomposition approach. The two-stage approach fails to find feasible solutions to medium-size problems. In terms of solution quality, BFS outperforms SA for small size problems, while SA generally performs better on problems with more transmission ratios.

All algorithms are tested on the *tremec dataset* and results are in Table 2. The time limit is 1 and 2 h for problems with 5 and 10 transmission ratios, respectively. Each algorithm is run 10 times for each problem. Results on this dataset confirms that the three-stage decomposition generally outperforms the two-stage. Among the algorithms, EDA is the best performing, mainly attributed to the larger number of solutions that can be evaluated in parallel.

8 Conclusion

In this paper, we consider the multi-speed gearbox synthesis problem formulated as a bi-level optimization problem. The inner task, or parameter optimization, is a non-convex continuous optimization problem, solved with a state-of-the-art solver. For the outer problem, we used two approaches to search over gearbox configurations: the two-stage decomposition approach searches over transmission ratio path assignments, while the three-stage decomposition approach first selects a sub-graph, and then performs the transmission ratio path assignment. We found that the latter consistently outperforms the first in all test problems and search algorithms. We investigated a variety of global search algorithms for solving the sub-tasks of the outer problem. While best-first-search usually performs well on small-size problems, the estimation of distribution algorithm produces better quality solutions for realistic instances. This work demonstrates the value of integrating methods from both the artificial intelligence and optimization fields applied to configuration design problems in mechanical engineering.

References

1. Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., Diehl, M.: CasADi: a software framework for nonlinear optimization and optimal control. Math. Program. Comput. **11**(1), 1–36 (2019). https://doi.org/10.1007/s12532-018-0139-4
2. Berx, K., Gadeyne, K., Dhadamus, M., Pipeleers, G., Pinte, G.: Model-based gearbox synthesis. In: Mechatronics Forum International Conference, pp. 599–605 (2014)
3. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
4. Buiga, O., Tudose, L.: Optimal mass minimization design of a two-stage coaxial helical speed reducer with genetic algorithms. Adv. Eng. Softw. **68**, 25–32 (2014)
5. Castro, M.P., Piacentini, C., Cire, A.A., Beck, J.C.: Relaxed decision diagrams for cost-optimal classical planning. In: Workshop HSDIP, ICAPS, pp. 50–58 (2018)
6. Chakrabarti, A., et al.: Computer-based design synthesis research: an overview. J. Comput. Inf. Sci. Eng. **11**(2), 021003 (2011)
7. Cheong, H., Ebrahimi, M., Butscher, A., Iorio, F.: Configuration design of mechanical assemblies using an estimation of distribution algorithm and constraint programming. In: 2019 IEEE Congress on Evolutionary Computation (CEC), pp. 2339–2346. IEEE (2019)
8. Chlamtáč, E., Dinitz, M., Makarychev, Y.: Minimizing the union: tight approximations for small set bipartite vertex expansion. In: Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 881–899. SIAM (2017)
9. Chomsky, N., Lightfoot, D.W.: Syntactic structures. Walter de Gruyter (1957)
10. Chong, T.H., Bae, I., Kubo, A.: Multiobjective optimal design of cylindrical gear pairs for the reduction of gear size and meshing vibration. JSME Int J., Ser. C **44**(1), 291–298 (2001)
11. Deb, K., Jain, S.: Multi-speed gearbox design using multi-objective evolutionary algorithms. J. Mech. Des. **125**(3), 609–619 (2003)

12. Dechter, R., Pearl, J.: Generalized best-first search strategies and the optimality of A. J. ACM (JACM) **32**(3), 505–536 (1985)
13. Eschenauer, H., Koski, J., Osyczka, A.: Multicriteria Design Optimization: Procedures and Applications. Springer, Heidelberg (2012)
14. Hauschild, M., Pelikan, M.: An introduction and survey of estimation of distribution algorithms. Swarm Evol. Comput. **1**(3), 111–128 (2011)
15. Huang, H.Z., Tian, Z.G., Zuo, M.J.: Multiobjective optimization of three-stage spur gear reduction units using interactive physical programming. J. Mech. Sci. Technol. **19**(5), 1080–1086 (2005)
16. Hwang, C.R.: Simulated annealing: theory and applications. Acta Applicandae Mathematicae **12**(1), 108–111 (1988)
17. Jain, P., Agogino, A.M.: Theory of design: an optimization perspective. Mech. Mach. Theory **25**(3), 287–303 (1990)
18. Königseder, C., Shea, K.: Comparing strategies for topologic and parametric rule application in automated computational design synthesis. J. Mech. Des. **138**(1), 011102 (2016)
19. Land, A., Doig, A.: An automatic method of solving discrete programming problems. Econometrica **28**(3), 497–520 (1960)
20. Li, X., Schmidt, L.: Grammar-based designer assistance tool for epicyclic gear trains. J. Mech. Des. **126**(5), 895–902 (2004)
21. Lin, Y.S., Shea, K., Johnson, A., Coultate, J., Pears, J.: A method and software tool for automated gearbox synthesis. In: ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, pp. 111–121. American Society of Mechanical Engineers Digital Collection (2009)
22. McKay, B.D., Piperno, A.: Practical graph isomorphism, II. J. Symb. Comput. **60**, 94–112 (2014)
23. Mittal, S., Frayman, F.: Towards a generic model of configuraton tasks. In: IJCAI, vol. 89, pp. 1395–1401. Citeseer (1989)
24. Mullins, S., Rinderle, J.R.: Grammatical approaches to engineering design, part i: an introduction and commentary. Res. Eng. Design **2**(3), 121–135 (1991)
25. Murthy, S.S., Addanki, S.: PROMPT: an innovative design tool. IBM Thomas J, Watson Research Division (1987)
26. Nadel, B.A., Lin, J.: Automobile transmission design as a constraint satisfaction problem: modelling the kinematic level. AI EDAM **5**(3), 137–171 (1991)
27. Osyczka, A.: An approach to multicriterion optimization problems for engineering design. Comput. Methods Appl. Mech. Eng. **15**(3), 309–333 (1978)
28. Pomrehn, L., Papalambros, P.: Discrete optimal design formulations with application to gear train design. J. Mech. Des. **117**(3), 419–424 (1995)
29. Prayoonrat, S., Walton, D.: Practical approach to optimum gear train design. Comput. Aided Des. **20**(2), 83–92 (1988)
30. Rai, P., Agrawal, A., Saini, M.L., Jodder, C., Barman, A.G.: Volume optimization of helical gear with profile shift using real coded genetic algorithm. Procedia Comput. Sci. **133**, 718–724 (2018)
31. Savsani, V., Rao, R., Vakharia, D.: Optimal weight design of a gear train using particle swarm optimization and simulated annealing algorithms. Mech. Mach. Theory **45**(3), 531–541 (2010)
32. Schmidt, L.C., Shetty, H., Chase, S.C.: A graph grammar approach for structure synthesis of mechanisms. J. Mech. Des. **122**(4), 371–376 (1999)
33. Sobieszczanski-Sobieski, J., Haftka, R.T.: Multidisciplinary aerospace design optimization: survey of recent developments. Struct. Optim. **14**(1), 1–23 (1997)

34. Swantner, A., Campbell, M.I.: Topological and parametric optimization of gear trains. Eng. Optim. **44**(11), 1351–1368 (2012)
35. TREMEC: Tremec. http://www.tremec.com. Accessed 28 Nov 2019
36. Tsai, L.W., Maki, E., Liu, T., Kapil, N.: The categorization of planetary gear trains for automatic transmissions according to kinematic topology. Technical report, SAE Technical Paper (1988)
37. Valiant, L.G.: The complexity of enumeration and reliability problems. SIAM J. Comput. **8**(3), 410–421 (1979)
38. Vinterbo, S.A.: A note on the hardness of the k-ambiguity problem. Technical Report DSG-T R-2002-006 (2002)
39. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. **106**(1), 25–57 (2006)
40. Wang, G.G., Shan, S.: Review of metamodeling techniques in support of engineering design optimization. J. Mech. Des. **129**(4), 370–380 (2006)
41. Wang, H., Wang, H.P.: Optimal engineering design of spur gear sets. Mech. Mach. Theory **29**(7), 1071–1080 (1994)
42. Wojnarowski, J., Kopeć, J., Zawiślak, S.: Gears and graphs. J. Theor. Appl. Mech. **44**(1), 139–162 (2006)
43. Yokota, T., Taguchi, T., Gen, M.: A solution method for optimal weight design problem of the gear using genetic algorithms. Comput. Ind. Eng. **35**(3–4), 523–526 (1998)
44. Zarefar, H., Muthukrishnan, S.: Computer-aided optimal design via modified adaptive random-search algorithm. Comput. Aided Des. **25**(4), 240–248 (1993)