Optimal Configuration Design: An Integrated Approach Using Grammars

L. C. Schmidt
Department of Mechanical Engineering,
The University of Maryland,
College Park, Maryland 20742-3035
lschmidt@eng.umd.edu

J. Cagan
Department of Mechanical Engineering,
Carnegie Mellon University,
Pittsburgh, Pennsylvania 15213-3890
cagan@cmu.edu

A computational approach to design that integrates conceptual design, configuration design, and component selection tasks overcomes some of the barriers to successful design automation. FFREADA is an implementation of a general design generation and optimization algorithm featuring hierarchical ordering of grammar-based design generation processes at different levels of abstraction. FFREADA is used to generate near-optimal hand-held drill power trains in a space exceeding 200 million designs that are not limited to any particular functional architecture or component configuration. Drill power train designs with values within 1 percent of the optimal solution are found in minutes by sampling 302,000 design states on average. Optimal configurations are found for drill power trains with three different torque requirements.

1 Introduction

Development of performance enhancing designer assistance tools requires research into strategies to capture good design alternatives. In mechanical design, optimization approaches are routinely used to set final parameters of components once basic design concepts, component configurations, and component selections are determined. Since optimization is a well-understood problem with powerful solution techniques, we are motivated to explore methods that formulate mechanical design tasks as optimization problems.

Sophisticated design tasks like generating overall solution ideas during conceptual design and determining the arrangement of components during configuration design are not amenable to traditional optimization for a variety of reasons. Designers may be unaware of all pertinent design objectives and constraints. Design specifications can begin as qualitative rather than quantitative goals, evading easy valuation modeling. Even clearly defined and well-understood design problems may have so many possible solutions that optimization resources are unavailable.

This work applies the FFREADA algorithm to the design of hand-held drill power trains. FFREADA uses a string grammar that arranges symbols representing power train components into a string, using grammar design rules that assure proper component operation. This FFREADA implementation integrates conceptual design, configuration design and component selection, formulating the design problem into one which is amenable to optimization. FFREADA's optimization strategy requires that every conceptual design be developed and converted into many alternative arrangements of specific components to fully investigate the potential of each concept. FFREADA is also used here to explore the power train design state space to learn about pertinent constraints and test objective valuation functions.

2 Configuration Design Problem

The configuration design task determines the way in which physical components with known inputs and outputs can be arranged to function properly. Configuration design requires selecting specific components from a pre-defined pool of "N" alternatives, each with "p" known connection port constraints (Mittal and Frayman, 1989). At its largest, the space of all possible designs is on the order of $\sqrt{(pN)}!$ and grows exponentially as new components are made available. One approach to reducing the size of the design space imposes a fixed functional architecture or the use of a key component on all designs, severely limiting the generation process. These were the primary strategies employed by the majority of configuration design systems studied from a knowledge-use perspective (Balkany et al., 1993). An alternative strategy is to use abstraction to decompose the design problem to search the solution space more efficiently (Kota and Ward, 1990; Snavely and Papalambros, 1993; Schmidt and Cagan, 1995). Earlier grammar-based work by the authors confirmed that FFREADA's abstraction level design approach was more efficient than designing from just a set of power train components. We believe that the abstraction-based approach imposed favorable topologies on the design state space, reducing search time (Schmidt, 1995). Finger and Rinderle (1989) used a specific type of grammar—bond graphs—to generate power transmission designs. This work differs from theirs in its use of an abstraction-based, multi-level grammar coupled with optimization techniques.

An additional barrier to the development of good configuration design models is the difficulty of evaluating a design prior to fixing component attributes (Finger and Dixon, 1989). Approaches to catalog selection design vary from expert systems applications to stochastic optimization approaches. A multi-objective catalog selection problem can be modeled as a task involving random design variables or problem uncertainties (Bradley and Agogino, 1991). The design task requires explicit calculations of the value of each feasible design solution available from the catalog of components. Another catalog selection approach abstracts component attributes into interval performance models to rule out complete classes of designs (Ward and Seering, 1989). FFREADA incorporates catalog selection as a separate level of design occurring concurrently with concept generation and configuration design.

3 FFREADA's Background

FFREADA, Function to Form REcursive Annealing Design Algorithm, is an implementation using grammars (Schmidt and Cagan, 1995, and Schmidt, 1995) of an abstraction-based model of mechanical design (Schmidt and Cagan, 1992) for design optimization and design space characterization. FFREADA, is a design generation and optimization algorithm featuring abstraction-level ordering of three grammar-based design generation processes (Fig. 1). FFREADA's code is written in the C

Contributed by the Design Automation Committee for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received Aug. 1996; revised Oct. 1997. Associate Technical Editor: A. R. Parkinson.

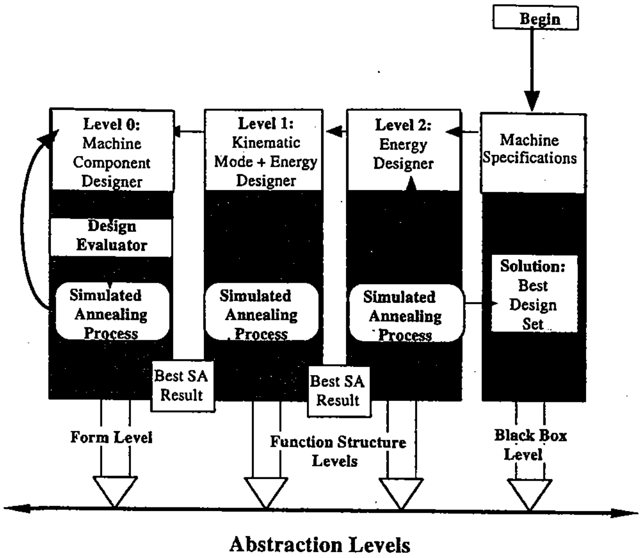


Fig. 1 FFREADA algorithm schema

programming language and runs on a DEC 3000/400 workstation.

To prepare the FFREADA algorithm for use, a designer chooses abstraction levels appropriate for the problem. Our selected levels are, from most to least abstract: energy (design Level 2), energy plus kinematic mode (Level 1), and form (Level 0). The designer also stocks FFREADA's libraries on each level of abstraction with entities that will create feasible designs using the established grammar rules.

3.1 FFREADA's Operation. The designer using FFREADA will input a set of machine specifications that will be passed to the energy level of abstraction as a pattern from which to create an abstract conceptual design of desired energy transformations. This design is too abstract to be evaluated, so it is passed down as a pattern to the kinematic mode + energy design level for conversion by grammar rules into a more detailed energy design. Again, it cannot be evaluated because metrics of interest (e.g., size, weight, performance, cost) cannot be evaluated until actual components are selected. So, this design is passed to the machine component designer which uses grammar rules to select actual components to be arranged according to the kinematic mode + energy pattern. This potential design can be evaluated. In fact, a complete simulated annealing process occurs on Level 0 to find the best component arrangement possible from the passed pattern. The value of the best design is passed up to Level 1 where it becomes one state in a concurrent annealing process to find the best kinematic mode + energy design possible from the given energy level pattern. In turn the results from simulated annealing on Level 1 become one state in the simulated annealing process concurrently occurring on Level 2 to find the best energy design for the power train. This recursive pattern of simulated annealing is diagrammed in Fig. 1.

3.2 Grammar-Based Design. A grammar is a formal mathematical construct consisting of a set of productions or rules, a set of symbols, and an initial symbol or symbol set. A grammar's rules manipulate an initial symbol into an arrangement of pre-defined symbols that creates a meaningful expression. In a language, the symbols are words and grammars arrange them into sentences. In this application, FFREADA's string grammar manipulates symbols representing power train functions and components into a series arrangement of symbols representing the actual power train design at three different levels of abstraction. A discussion of recent grammar applications in mechanical design research can be found in Schmidt and Cagan (1996).

3.3 Recursive Simulated Annealing Optimization. Simulated annealing (SA) is a stochastic optimization technique introduced by Kirkpatrick et al. (1983). The SA algorithm begins at an initial design state of known value. The algorithm generates a new design state by randomly modifying the current design. If the new state's value is better than the previous one, it is accepted as the current solution; if it is worse, it is accepted with a decreasing probability (P_{accept}) given by:

$$P_{\text{accept}} = e^{-\Delta C/T}, \tag{1}$$

where ΔC is the difference in the values of the states, and T is an algorithm parameter called 'temperature' that decreases throughout the optimization process. At the outset, many inferior solutions are accepted, resulting in "hill-climbing" or nearly "random exploration" behavior in order to find promising regions of the design space. As the temperature continues to decrease, algorithm behavior resembles downhill search because virtually no inferior steps are accepted. This allows the SA algorithm to converge to a local optimum in the current region of the design space, which is believed to hold good solutions.

4 Hand-Held Drill Power Train Design Problem

Here FFREADA is employed to assist a power tool maker considering creating a new cordless drill product line. Cordless, hand-held power drills are commonly used by homeowners to drive wood or metal screws. The drill operates by pulling a trigger which sends a signal to activate the power supply. The output of the drill is high speed rotation of the chuck and drill bit.

FFREADA uses an entity representation system for design that follows the Pahl and Beitz (1988) energy, material, and signal flow machine representation scheme.

Specifications for a power train are:

Drill Power Train Specifications:

Inputs: energy—no flow, material—no flow, signal—yes, a triggering action;

Output energy—continuous rotational energy, material—bit, signal—no flow.

This FFREADA implementation has limitations. Issues of ergonomics, motor performance, stress evaluation of gear selections, power transmission losses, to name a few, are not dealt with here. Instead, we focus on demonstrating the benefits of FFREADA's abstraction-grammar design process and recursive simulated annealing strategy.

4.1 Function and Form Entity Representation. FFREADA uses entity libraries existing on each of three levels of abstraction defined for this problem. Each entity is represented as $S_{i,j}$, where "i" is the entity identification number and "j" is the entity's level of abstraction. Figure 2 displays a FFREADA drill power train design represented as a tree spanning the three levels of abstraction. The full design includes interpretations of the design at each level of abstraction, which serve as patterns for designing on the next lower level of abstraction.

FFREADA's form entity library, used for designing on abstraction Level 0, holds drill power train components: power supplies, motors, shafts, and spur gears. The symbol $S_{3,0}$ represents one particular power supply. Entity $S_{3,0}$ has no energy or material inputs but requires a signal input to produce a continuous electrical energy output. Abstraction Level 1 designs using general energy and kinematic mode details to create classes of components for use in catalog design. On this level, a direct current power supply like $S_{3,0}$, belongs to the entity class $S_{52,1}$,

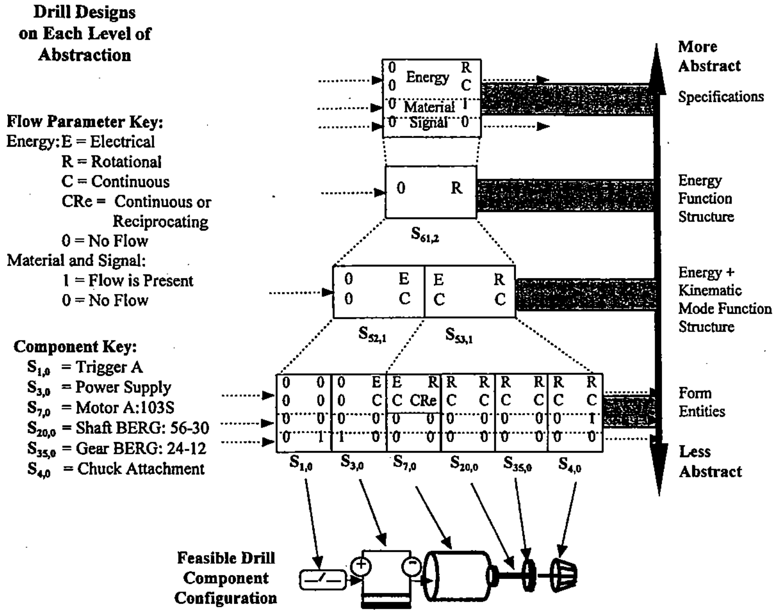


Fig. 2 Typical FFREADA drill power train design

"$0 \rightarrow E$ $0 \rightarrow C$," modeling no energy input with electrical energy (E) output on a continuous (C) basis. On abstraction Level 2, components are abstracted into entities described only by the energy transformations they provide. Here, entity $S_{58,2}$ represents any power supply that takes no energy input (only a signal) and outputs electrical energy in any form. Level 0 entities are combined to create the energy transformation conceptual design for a power train. Details on FFREADA's entity representation system and complete library listings are included in the Appendix.

4.2 Grammar-Based Design Process. FFREADA uses context-sensitive, abstraction-grammar rules. A typical rule schema on any level "m" of the abstraction grammar is the following:

$$S_{w,m}X_m \rightarrow S_{w,m}S_{x,m}X_m, \text{ where } \exists\, A_{x,m} \subset S_{x,m} \text{ such that } A_{x,m} = P_{w,m} \subset S_{w,m}. \quad (2)$$

Here, X_m is a nonterminal symbol of the grammar which can not appear in a final string. $S_{w,m}$ and $S_{x,m}$ are terminal symbols and represent function structure entities on level "m" with activation ($A_{x,m}$) and production parameter sets ($P_{w,m}$). Activation parameters are those energy inputs required to cause the entity to function. Production parameters are the energy outputs from the functioning entity. Feasible designs require that the activation parameters of one entity match or are compatible with the product parameters of the entity preceding it in a series arrangement. In addition to energy parameters, material and signal parameters also conform to the same domino matching strategy. Grammar rules referencing entity information enforce the matching strategy thereby ensuring the creation of only feasible designs. Form level entity connectivity parameters must also be compatible for grammar rule applications to be valid (e.g., pitches on mating gears must be equivalent).

Implicit in FFREADA's designing is the generation of a power train configuration. It is generalized from the design on the form level of abstraction. FFREADA generated this design string for the power train of Fig. 2.:

$$S_{1,0}S_{3,0}S_{52,1}S_{7,0}S_{20,0}S_{35,0}S_{4,0}S_{53,1}S_{61,2}. \quad (3)$$

The power train configuration generalized from the highlighted form level entities of this string is as follows:

$$\text{trigger—power supply—motor—shaft—gear—chuck.} \quad (4)$$

This power train satisfies designer specifications by delivering rotational energy of 0.53 J at 2000 rpms on a continuous basis for a cost of \$246.11[1] and a weight of 6.4 kg. This represents a feasible solution, yet, it is relatively expensive and includes a superfluous shaft and gear assembly. FFREADA's simulated annealing guides the design process so that feasible but poor designs are ultimately discarded.

4.3 Size of Drill Power Train Design Problem. The size of the space of designs articulated by a grammar provides a baseline for comparison to the number of design generations required for algorithm convergence. Using combinatorics, the 6-component power train configuration of Fig. 2 is the only feasible solution that exists, but it can be further detailed into 2,475 different component level designs. Valid 7-component configurations feature either a "gear-gear" or a "gear-shaft" assembly located behind the drill chuck attachment. There are 32,175 7-component design options. At the 9- and 10-component configuration sizes there are about 10 and 200 million designs, respectively.

As the component design limit increases, more power trains can be designed but many of these are ineffective with many extra gear-pairs and shafts. During FFREADA's annealing runs a 250-component limit is applied to the drill designs because it actually saves run time with this implementation[2]. Conservatively estimating a 10-fold increase in design solutions with each 1-component increase, a 250-component limit defines a design space exceeding 10^{249} states.

[1] Drill costs are high because they are developed by adding the cost to purchase one of each component in the design. No economies of scale are assumed in the component costs. These cost figures are adequate for evaluating the relative costs of drill designs.

[2] In the present implementation, the high component limit on designs reduces the running time of the algorithm because the length check is applied after the design is generated. Feasible but long designs take time to generate only to be discarded. One of FFREADA's 200,000 design generations for the cost-with-penalty objective function required 10.1 minutes run time on a DEC 3000/400 with a 250-component limit and 46.6 minutes with a 10-component limit. FFREADA's annealing algorithm will ultimately reject long designs in favor of shorter ones, so a low-component limit is unnecessary.

Table 1 Random design generation results for simple valuation functions

Run #	Valuation Function	Minimum Value	Maximum Value	Average Value	Sample Standard Deviation
1	Number of components	5.0	10.0	7.44	1.67
2	Weight (kg)	6.4	135	27.1	17.4
3	Torque (J)	0.014	67.8	0.99	1.51
4	Cost ($)	114	1,045	272	123
5	Cost/Torque ($/J)	8.19	10,878	473	369
6	Cost ($) with Penalty* for Drills of less than 1.356 J (1 ft-lb) Torque 10-component limit on designs	124.74	5,247.05	1,124.70	509.46
7	Cost ($) with Penalty* for Drills of less than 1.356 J(1 ft-lb) Torque 250-component limit on designs	125.79	32,654.65	3,907.87	3,742.99

*The cost penalty is a 400% cost mark-up on the drill.

5 FFREADA Results

FFREADA's implementation identified good drill power train designs by optimizing quantifiable performance characteristics. FFREADA's design generation and evaluation process was also used to explore the design state space by creating probability frequency histograms of the values of randomly generated designs.

5.1 Drill Power Train Design State Space Survey. Several FFREADA random design generation runs of 200,000 designs each were done to survey the power train design state space. A variety of valuation functions were investigated. Table 1 data describes performance characteristics that dominate power train valuation, characteristics suitable for commercial drill design, and demonstrates how the design state space changes when mapped by different objective functions.

Motor selection dominates power train performance in terms of: cost, weight, and torque produced. FFREADA's grammar generates feasible power trains with torque at motor spindle levels which range from 0.39 to 0.92 J, and includes one motor at 4.2 J of torque. Power train designs yielding between 0.81 and 4.2 J torque are not readily found by random design generation, as is evidenced by the drill torque histogram of Fig. 3.

To produce designer-specified torque, a minimum torque constraint could be imposed, but we chose instead to apply a penalty to the evaluation function for designs with insufficient torque to avoid limiting the grammar's design freedom. Since we have defined good power train designs as low-cost, a cost objective valuation function is needed. Possible valuation functions are tested with results displayed in Table 1 and in the histograms of Fig. 4.

The cost-per-torque ($/J) histogram included in Fig. 4 is interesting but unbounded and, therefore, inadequate. We illustrate the ability of FFREADA's search strategy to control the design generation process by looking at a "good" design found using this evaluation function. The design was valued at $0.27 per J of torque, had 44 components, generated 339 J of torque, and contained 14 shafts separated by various arrangements of gear pairs.

The valuation function selected for power train design was the cost-with-penalty function, used in Run 6 and Run 7 of Table 1. It increases the cost of power trains producing less than 1.356 J (1 ft-lb) of torque by 400 percent. A histogram of designs evaluated by this function is included in Fig. 4. The histogram peak at about $530 represents the penalty barrier for the low-torque power trains.

5.2 Results from Annealing Optimization Mode. FFREADA was applied in optimization mode using the cost-with-penalty objective function and a relaxed component limit of 250. The annealing schedule on each level of abstraction was a simple derivative of the adaptive annealing schedule of Huang et al. (1986). Rather than a probabilistic approach to determining the achievement of convergence, FFREADA used a consecutive-rejection strategy (50-, 10-, and 10-rejection limits on

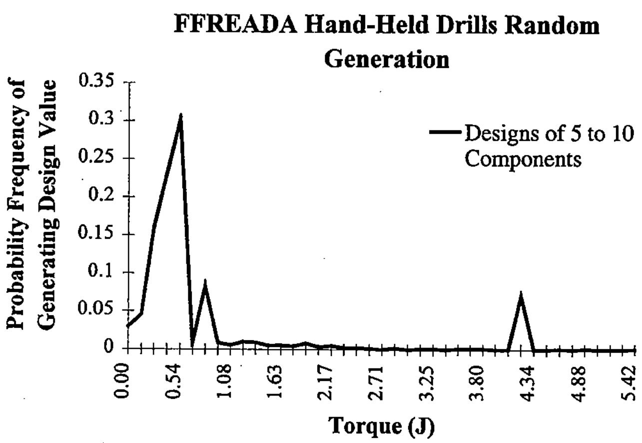


Fig. 3 Torque probability frequency histogram

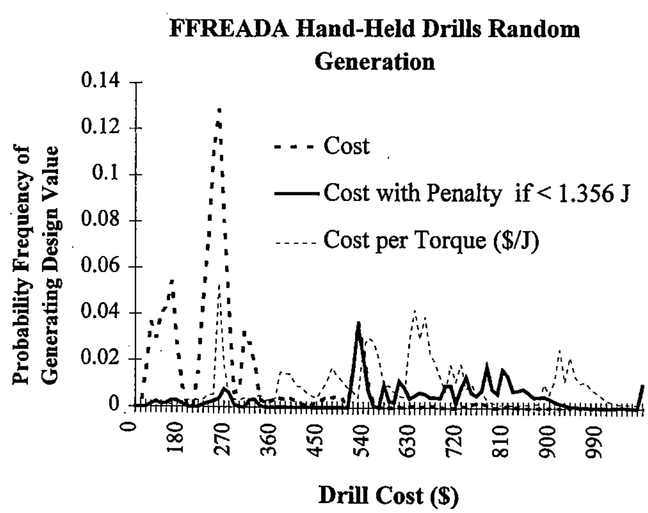
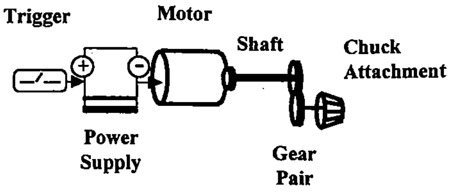
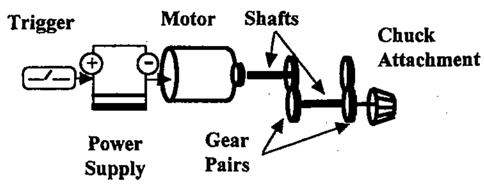


Fig. 4 Various cost-based histograms



(a) Optimal configuration for 1.36 J of torque at minimum cost



(b) Optimal configuration for 2.71 and 4.07 J of torque at minimum cost

Fig. 5 FFREADA's optimal design configurations

Table 2 Recursive annealing results vs down-hill quench results

Run	FFREADA Recursive Annealing		FFREADA Down-hill Quench		FFREADA Random Design Generation
	Run Result Cost ($)	Run Minimum Cost ($)	Run Result Cost ($)	Run Minimum Cost ($)	Run Minimum Cost ($)
1	125.64	125.64	126.97	126.97	125.79
2	130.15	125.16	127.10	127.10	124.82
3	126.93	124.74	126.21	126.21	125.61
4	126.04	124.82	127.14	127.14	125.24
5	126.04	125.61	126.97	126.97	124.74
6	124.74	124.74	129.13	129.13	125.66
7	125.66	125.61	125.73	125.73	125.24
8	126.58	124.82	126.17	126.17	124.82
9	126.78	125.24	125.95	125.95	125.73
10	126.26	125.87	125.96	125.96	125.24
Average	126.48	125.23	126.73	126.73	125.29
Standard Deviation	1.36	0.41	0.95	0.95	0.38
Average # Form Level Designs	301,391		28,860		302,000 Set by Designer

Levels 2, 1, and 0, respectively) and an overall 1000-design iteration limit at any temperature on each Level.

5.2.1 Design Scenario #1: 1.36 J (1 ft-lb) Torque Requirement. The optimization results from 10 FFREADA runs appear in the left columns of Table 2[3]. FFREADA converged to the globally optimal minimum cost ($124.74) design for a power train generating at least 1.36 J of torque in Run 6. The design string is:

$$S_{1,0}S_{3,0}S_{10,0}S_{16,0}S_{45,0}S_{47,0}S_{4,0}S_{50,1}S_{61,2} \qquad (5)$$

and the optimal configuration design is displayed in Fig. 5(*a*). Three, independent, random generation runs of 2,000,000 designs each confirm this design as the global minimum.

FFREADA's program output is as follows:

Design Entity Parameters

Entity Name	Energy	Material	Signal
Level 2:	$0 \rightarrow R$	N/A	N/A
Level 1:	$0 \rightarrow R$ $0 \rightarrow C$	N/A	N/A
Level 0:			
Trigger B, $S_{1,0}$	$0 \rightarrow 0$ $0 \rightarrow 0$	$0 \rightarrow 0$	$0 \rightarrow 1$
Power supply, $S_{3,0}$	$0 \rightarrow E$ $0 \rightarrow C$	$0 \rightarrow 0$	$1 \rightarrow 0$
Motor (HL:2100), $S_{10,0}$	$E \rightarrow R$ $C \rightarrow CRe$	$0 \rightarrow 0$	$0 \rightarrow 0$
Shaft (BERG: 54–25), $S_{16,0}$	$R \rightarrow R$ $C \rightarrow C$	$0 \rightarrow 0$	$0 \rightarrow 0$
Gear (BERG: 64–12), $S_{45,0}$	$R \rightarrow R$ $C \rightarrow C$	$0 \rightarrow 0$	$0 \rightarrow 0$
Gear (BERG: 64–30), $S_{47,0}$	$R \rightarrow R$ $C \rightarrow C$	$0 \rightarrow 0$	$0 \rightarrow 0$
Chuck, $S_{4,0}$	$R \rightarrow R$ $C \rightarrow C$	$0 \rightarrow 1$	$0 \rightarrow 0$
Design cost:	$124.74		
Design torque:	1.38 J (130 Watts Maximum Power)		
# Form level designs generated in run:	271,611		
Run time:	3.25 Minutes		

FFREADA's recursive annealing results in Table 2 display convergence to designs with costs within 1.010 percent of the minimum generated in each run and within 1.014 percent of the global minimum of $124.74. The average number of form level designs generated per run is 301,391. This represents less than 0.15 percent of a 200 million design state space resulting from a 10-component limit and an insignificant portion of design states in the expanded 250-component design space.

5.2.2 Design Scenario #2: 2.71 J (2 ft-lb) Torque Requirement. FFREADA application to design a minimum cost power train, generating at least 2.712 J of torque, led to the two-gear-pair, optimal configuration shown in Fig. 5(*b*). Ten FFREADA runs using the previous convergence limits yielded an average cost power train design of $145.21 in an average of 4.8 minutes, from an average of 141,992 form level designs. The optimal configuration was found in each case, however the minimum cost solution found is $138.37 and the optimal solution has a cost of $136.48. Repeating the process with stricter consecutive rejection limits (300-, 250- and 25-for design abstraction levels 2, 1, and 0, respectively) increases the run time and reduces the average solution cost from $145.21 to $138.89, which lies within 1.8 percent of the optimal solution.

5.2.3 Design Scenario #3: 4.07 J (3 ft-lb) Torque Requirement. We changed the problem to require a goal torque of 4.07 J and applied FFREADA with the strict convergence criteria. The optimal configuration was the same as for the 2.71 J solution. The optimal solution found manually using knowledge of the best configuration and the component library cost $137.28. FFREADA's average design solution had a $139.51 cost, within 1.6 percent of the optimum, with an average of 2,599,407 form level designs generated for convergence. FFREADA generates designs at a rate of about 80,000 per minute.

6 Discussion

6.1 Optimal Configuration Design and Optimal Catalog Selection Results. In each power train design scenario, FFREADA converged to designs with the optimal configuration [Figs. 5(*a*) and 5(*b*)] and an optimal or near-optimal set of power train components in the optimal configuration. FFREADA proved to be an effective tool for optimal configuration design but there is room to improve the integrated catalog selection design task. For small component libraries, it might be more efficient to use a deterministic approach to the selection of components. Nothing in FFREADA's design model prohibits applying more sophisticated techniques or mixing search techniques. On the contrary, the hierarchical approach to the design problem makes it possible to apply different strategies on each level of abstraction.

6.2 Goodness of Recursive Annealing Strategy. FFREADA's hierarchical optimization strategy provided an opportunity to experiment with different methods. Table 2 com-

[3] The remainder of Table 2 includes results obtained using other search strategies and is discussed in Section 6.2.

pares recursive annealing results to results obtained using a down-hill search strategy (here called a quench) and long random design generation runs. In the quench, only design states with values lower than the current state are accepted as the new current design state. Table 2's quench results indicate shorter runs and guaranteed convergence to the run minimum. Overall results were not quite as good as FFREADA's recursive annealing results because the exploration of the design space is cut short due to early convergence. Quench searches were ultimately trapped in local minima giving witness to the ill-behaved nature of the design space. Additional exploration required for convergence under the simulated annealing strategy becomes more valuable as the design valuation space becomes more discontinuous.

Table 2 also displays the minimum cost power train from 10 random design generation runs. This set averages a run minimum similar to the recursive annealing strategy and found the optimal power train design once. The real significance of the comparison to the random generation results is that FFREADA's results came without any foreknowledge of the design state space. The random generation results were found only after setting a 302,000 run generation limit. This run limit was determined from examining FFREADA's good results. No other information exists to suggest how long to randomly search for good results. For example, three independent runs of 50,000 design generations each found best designs valued at $126.26, $125.64, and $126.04, but never the optimal solution. FFREADA's annealing strategy yielded the best design results achievable with the least amount of a priori knowledge.

6.3 Comparison to Commercial Drill Power Trains. FFREADA designed without power generation goals using only a cost-minimizing goal. As a result, FFREADA's near-optimal power train designs used a low-cost, $\frac{1}{5}$ horsepower motor and can't compare in performance to commercial drill designs. However, the optimal component configurations are similar to commercial drills.

- Black & Decker's cordless (4.8V), reversible, variable-speed (to 450 RPM), $\frac{3}{8}''$ drill (Model CD1000) features the two gear-pair configuration suggested as optimal for 2.71 and 4.07 J torque drills.
- DeWalt® corded industrial drills use a single gear reduction for drills with maximum power output at 350 W and a double reduction for maximum power over 400 W.

Increasing numbers of gear reductions as the power requirements increase is consistent with the designs generated by FFREADA. Two other gear configurations are seen on the market: triple-reduction (corded Craftsman Model 91027), and a five-gear planetary transmission (cordless DeWalt's® DW900 VSR series). FFREADA's current grammar will not design these power trains; however, new grammar rules could be added to expand the language of the grammar.

6.4 Comparison to Commercial Design Processes. We have assumed that a designer used FFREADA to generate power trains from an inventory of components on hand. The designer's goal was to learn the types of configurations necessary to meet stated performance requirements. An actual drill manufacturer, like Black & Decker, would likely custom design and build a suitable motor based on power requirements rather than choose from a motor catalog.

The power trains generated in this work considered only a few of the possible objectives of a manufacturer. Furthermore, the drill components and their abstracted function entities are limited in their expression of behavior. More powerful design grammars for drill power trains would require representations and rules that express geometric constraints and power train behaviors like changing directions of rotation and axes of orientation.

Conclusions

A designer can make more informed decisions on the potential of conceptual or configuration designs when using a design tool that evaluates downstream designs from each alternative. FFREADA's integrated abstraction-grammar based design generation approach is successful in generating optimal design configurations for three different drill power train design problems and found near optimal complete solutions in each case. FFREADA's success is attributed to its ability to convert candidate configurations into real components, enabling accurate design evaluations. FFREADA's recursive simulated annealing optimization strategy is also powerful enough to focus the search to converge in minutes. While drill power trains are a well-understood design area, tailoring this approach for new classes of designs will allow designers to explore the space of possible solutions and develop insight about the design problem. Current work continues on implementing other grammars to explore more complex mechanical design problems. For example, simple cart designs have been developed using a graph grammar for the design process (Schmidt and Cagan, 1997), allowing for complex and non-serial component arrangements.

Acknowledgments

The authors acknowledge Dr. Ramesh Krishnamurti of Carnegie Mellon University for discussions on grammar development. Linda Schmidt gratefully acknowledges the support of the National Defense Science and Engineering Graduate (NDSEG) Fellowship Program and the National Science Foundation grant DMI 96-3470. The authors also wish to acknowledge the National Science Foundation for partial funding of this research under grants #DDM-9258090 and #DDM-9301096.

References

Balkany, A., Birmingham, W. P., and Tommelein, I. D., 1993, "An Analysis of Several Configuration Design Systems," *AI EDAM*, Vol. 7, No. 1, pp. 1–17.

Bradley, S., and Agogino, A., 1991, "An Intelligent Real Time Design Methodology for Catalog Selection," *ASME '91 Design Theory and Methods*, ASME DE-31, pp. 201–208.

Finger, S. and Dixon, J. R., 1989, "A Review of Research in Mechanical Engineering Design. Part I: Descriptive, Prescriptive, and Computer-Based Models of Design Processes," *Research in Engineering Design*, 1, pp. 51–67.

Finger, S. and J. Rinderle, 1989, "Transformational Approach to Mechanical Design Using a Bond Graph Grammar," *1st International Conference on Design Theory and Methodology*, ASME DE-17, pp. 107–116.

Huang, M. D., Romeo, F., and Sangiovanni-Vincentelli, A., 1986, "An Efficient General Cooling Schedule for Simulated Annealing," *Proceedings of the IEEE Inter-national Conference on Computer-Aided Design (ICCAD-86)*, Santa Clara, CA, November 11–13.

Kirkpatrick, S., Gelatt, Jr., C. D., and Vecchi, M. P., 1983, "Optimization by Simulated Annealing," *Science*, 220(4598), pp. 671–679.

Kota, S., and Ward, A. C., 1990, "Functions, Structures, and Constraints in Conceptual Design," *Proceedings of the 2nd International Conference on Design Theory and Methodology*, Chicago, IL, DE-27, pp. 239–250.

Mittal, S., and Frayman, F., 1989, "Towards a Generic Model of Configuration Tasks," *IJCAI-89, Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, August 20–25, Vol. 2, pp. 1395–1401.

Pahl, G., and Beitz, W., 1988, *Engineering Design—A Systematic Approach*, Springer-Verlag, New York.

Schmidt, L., 1995, "An Implementation Using Grammars of an Abstraction-Based Model of Mechanical Design for Design Optimization and Design Space Characterization," Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA.

Schmidt, L., and Cagan, J., 1992, "A Recursive Shape Annealing Approach to Machine Design," *Preprints: The Second International Round-Table Conference on Computational Models of Creative Design*, Heron Island, Queensland, December 7–11, pp. 145–171.

Schmidt, L., and Cagan, J., 1995, "Recursive Annealing: A Computational Model for Machine Design," *Research in Engineering Design*, Vol. 7, pp. 102–125.

Schmidt, L., and Cagan, J., 1996, "Grammars for Machine Design," J. S. Gero and F. Sudweeks, eds, *Artificial Intelligence in Design 96*, Kulwer Academic Publishers, pp. 325–344.

Schmidt, L., and Cagan, J., 1997, "GGREADA: A Graph Grammar-Based Machine Design Algorithm," *Research in Engineering Design*, Vol. 9, pp. 195–213.

Snavely, G. L., and Papalambros, P. Y., 1993, "Abstraction As a Configuration Design Methodology," *Proceedings of the 19th ASME Design Automation Conference*, Albuquerque, NM, September 19–22, DE-65(1), pp. 297–305.

Ward, A., and W. Seering, 1989, "The Performance of a Mechanical Design Compiler," MIT AI Memo #1084.

APPENDIX A

FFREADA'S Drill Power Train Design Entity Libraries

Entities $S_{i,j}$ are modeled by activation (A) and production (P) energy parameters, pattern information (M), and knowledge used to express performance (K) as $S_{i,j} = (A \cup P \cup M \cup K)$, where i is an entity identification number and j is the entity's level of abstraction. The activation and production parameters of entities for drill power train design are arranged in three new parameter sets, energy (En), material (Ma), and signal (Sg) flow parameters. Functional pattern information remains in set M and the entity knowledge set, K, is tailored to include parameters necessary to evaluate power train designs.

$S_{i,j} = (En_{i,j} \cup Ma_{i,j} \cup Sg_{i,j}) \cup M \cup K$, with:

En = $\{A \cup P\}$, energy flow, where $A = \{a_2, a_1\}$, signifies the type of entity activation energy and $P = \{p_2, p_1\}$, signifies the type of energy produced by or output from the entity. Parameter subscripts indicate the level of abstraction with which the characteristic is associated. For example, Level 2 is defined by energy type parameters a_2 and p_2. Parameters $a_2, p_2 \in$ {rotational (R), translational (T), electrical (E), none (0)}; and, $a_1, p_1 \in$ {continuous (C), reciprocating (Re), both behaviors (CRe), none (0)} where 0 indicates no energy flow.

Ma = $\{c_0, r_0\}$, material flow, $b_0, c_0, q_0, r_0 \in \{0, 1\}$, where 0 indicates no flow and 1 indicates flow; and

Sg = $\{b_0, q_0\}$, signal flow.

M = $\{S_{i,(j+1)}\}$, an entity on the next higher level of abstraction that entity $S_{i,j}$, alone or in series with other jth-level entities, fulfills.

K = $\{k_i\}$, $i = 1, 2, \ldots, 9$, entity knowledge parameters where:

k_1 = name,
k_2 = input connector type,
k_3 = output connector type,
k_4 = torque (J) produced in operation,
k_5 = angular velocity (rpms) produced in operation,
k_6 = number of teeth (for gears),
k_7 = measure (e.g., length for shafts, outer diameter for gears (cm)),
k_8 = weight (g) aluminum at 13.2 g/cm^3, 0.3175 cm thickness, Shaft weights are calculated assuming

Table A1 FFREADA's drill power train library function entities

$S_{i,j}$	Name	Inputs		Outputs	
i,j	k_1	a_2	a_1	p_2	p_1
56,2	R→E	R	0	E	0
57,2	R→0	R	0	0	0
58,2	0→E	0	0	E	0
59,2	E→R	E	0	R	0
60,2	E→0	E	0	0	0
61,2	0→R	0	0	R	0
50,1	0→R 0→C	0	0	R	C
51,1	R→R C→C	R	C	R	C
52,1	0→E 0→C	0	0	E	C
53,1	E→R C→C	E	C	R	C
54,1	R→E C→C	R	C	E	C
55,1	R→0 CRe→0	R	CRe	0	0

Table A2 Generic machine component parameters

Name	Input Connector Type	Output Connector Type	Input Parameters				Output Parameters			
k_1	k_2	k_3	a_2	a_1	b_0	c_0	p_2	p_1	q_0	r_0
trigger B	j	b	0	0	0	0	0	0	1	0
trigger A	a	b	R	C	0	0	0	0	1	0
power supply	b	c	0	0	1	0	E	C	0	0
chuck	k	adefghik	R	C	0	0	R	C	0	1
motors	c	ad	E	C	0	0	R	CRe	0	0
shafts	d	acefghik	R	C	0	0	R	C	0	0
gears	See Table A4	See Table A4	R	C	0	0	R	C	0	0

Table A3 Specific component parameters for non-gear components

$S_{i,0}$	Name	Knowledge Parameters				
i	k_1	k_4 J	k_5 rpms	k_7 cm	k_8 kg	k_9 $
1	trigger B				0.22	1.00
2	trigger A				0.22	1.00
3	power supply				2.20	1.00
4	chuck				0.68	1.00
	MOTORS	k_4 J	k_5 rpms	k_7 cm	k_8 kg	k_9 $
5	A:101SM	.64	2200		6.2	245.00
6	A:55SM	.39	1500		3.3	230.00
7	A:103S	.53	2000		5.1	235.00
8	LMP:3003E	.42	2000		6.6	135.00
9	A:360AC	4.24	2000		6.6	510.00
10	HL:2100M	.56	2200		5.5	110.00
11	HZ:2100MX	1.73	2000		11	295.00
12	B:10-ABP-0	.57	1725		6.4	150.00
13	B:74-ABP-0	.39	3450		6.6	224.00
14	B:VL-3501	.64	2850		6.8	245.00
15	B:KL-3235	.53	2850		6.4	210.00
	SHAFTS	k_4 J	k_5 rpms	k_7 cm	k_8 g	k_9 $
16	BERG:54-25			6.4	79	1.06
17	BERG:54-30			7.6	93	1.14
18	BERG:54-35			8.9	108	1.14
19	BERG:56-25			6.4	143	1.98
20	BERG:56-30			7.6	172	2.11
21	BERG:56-35			8.9	201	2.11
22	BERG:28-25			6.4	201	2.77
23	BERG:28-30			7.6	254	3.12
24	BERG:28-35			8.9	333	3.52

Table A4 Specific gear parameters

$S_{i,0}$	Gear Name	Knowledge Parameters			
i	k_1	k_6 #	k_7 cm	k_8 g	k_9 $
25	BERG:48-12	12	.76	1.90	5.23
26	BERG:48-14	14	.89	2.64	5.40
27	BERG:48-30	30	1.73	9.70	5.87
28	BERG:48-44	44	2.44	21.2	6.59
29	BERG:48-48	48	2.64	21.2	6.80
30	BERG:32-12	12	1.12	4.19	5.01
31	BERG:32-14	14	1.27	5.29	5.93
32	BERG:32-30	30	2.54	44.1	6.57
33	BERG:32-44	44	3.73	44.1	6.95
34	BERG:32-48	48	4.06	55.1	6.97
35	BERG:24-12	12	1.47	7.06	6.00
36	BERG:24-14	14	1.70	9.48	6.26
37	BERG:24-30	30	3.56	41.9	7.01
38	BERG:24-44	44	4.83	77.2	7.77
39	BERG:24-48	48	5.33	92.6	7.77
40	BERG:16-12	12	2.24	16.3	7.69
41	BERG:16-14	14	2.54	21.2	8.40
42	BERG:16-30	30	5.08	83.8	10.33
43	BERG:16-44	44	6.93	158	12.40
44	BERG:16-48	48	7.94	207	12.49
45	BERG:64-12	12	.81	2.21	5.01
46	BERG:64-14	14	.91	2.87	5.21
47	BERG:64-30	30	1.27	5.29	5.67
48	BERG:64-44	44	1.83	11.0	6.46
49	BERG:64-48	48	1.98	13.2	6.68

0.6325 cm diameter and assuming a material of stainless steel (weight of 38.2 g/cm^3).

k_9 = cost ($). Single unit entity purchase costs are obtained from actual component suppliers. No bulk discounts are assumed.

Knowledge parameters for the form level power train components are obtained from various component manufacturers' catalogs. FFREADA's library contains triggers (2), power supplies (1), chuck attachments (1), motors (11), shafts (9), and gears (25), for a total of 49 form level components.

Tables of the function and form entities used for the power train design problem are included here. The form level power train component information was taken from the following catalogs:

- Berg Precision Mechanical Components (B6A, B8, and M-90),
- Stock Drive Products Handbook of Small Standardized Components (Master catalog 757),
- Aerotech Motion Control Product Guide 1994,
- Linear Motion Products Truly Affordable Automation Catalog (H800),
- Baldor Motors and Drives 501 1994 Stock Product Catalog, and
- Baldor Small Motors and Gear Motors Stock Catalog (501 CK).

The function entities are described in Table A1. All b_0, c_0, q_0, and r_0 parameters for function entities are equal to 0. FFREADA's function and form entities are described in Tables A2 and A3. Gear data is summarized in Table A4. The gear names include information about the pitch and number of teeth. BERG:48-12, is a 48 pitch, 12 toothed gear. Gear knowledge parameters k_2 and k_3 are not listed here but would hold connectivity information incorporating the pitch requirements. For example, BERG:48-12 has parameters $k_2 = e$ and $k_3 = acdek$. This gear can be connected to any gear of pitch 48 and also to motors, shafts, and a chuck attachment.

(Contents continued)