Tutorial: Applying Evolutionary Computation in Industrial Practice

Erik Goodman, Professor & Director
BEACON Center for the Study of Evolution in Action
(An NSF Science & Technology Center)

IEEE WCCI 2016
Vancouver, BC

IEEE WCCI 2016
Vancouver Canada

BEACON
An NSF Center for the Study of Evolution in Action

My Experience in Applying EC in Industry

Co-founder and VP Tech., Red Cedar Technology, Inc. (now sells $ millions/year of engineering design optimization software to industry) ***(I've left the company**—it's now owned by Siemens PLM, the Unigraphics NX folks)*

Director, BEACON Center, an NSF Science and Technology Center funded at $47.5M to study evolution in action, including ~20 faculty members studying computational evolution/EC; companies sponsor projects

Before BEACON, co-directed Genetic Algorithms Research & Applications Group (GARAGe); did research for industry

Former Director, MSU Manufacturing Research Consortium (Ford, GM, others were members)

Former Director, A. H. Case Center for Computer-Aided Engineering and Manufacturing, MSU

PI of Research under Contracts to General Dynamics, Ford, General Motors, Chrysler, Motor Wheel, etc.

Overview

We'll look at Applying EC in Industrial Practice from:

- **Industry point of view**
  - **4 ways of accessing EC**
- **Academic point of view**
  - **6 ways of accessing industry**
- **Case study for prospective EC entrepreneurs**

The Fundamental Conflict— (Oversimplified)

What We Do in Academia:

- Innovative new algorithms that excel in performance on (benchmark) problems as similar to industrial problems as we can find/create

What Industry Needs:

- “Packaged” and well-supported systems that will still be supported and improved in 5 years.
- Software with easy interfaces with the software already being used.
- Using must NOT require expert decision-making/tuning to apply to company’s problems.
- OR *contracting with someone else* to solve the whole problem, and within the COMPANY’s timeframe.

First, Let's Look from the Company's Point of View

Even academics need to know:

- What are industrial "customers" looking for
- What are the problems in relating to them as an academic

Industry Viewpoint: How Can a Company Access the Latest in EC Technology?

- Plan I: Track progress in the field and seek a research contract/grant with a faculty member/center at a leading institution/consortium/lab
- Plan II: Seek out a consulting company with a good reputation for solving the kind of problems the company faces
- Plan III: Find a solid company that develops/sells commercial software; license the software; get company engineers trained
- Plan IV: Track field & develop software in-house

How Can a Company Access the Latest in EC Technology?

Plan I: Track progress in the field and seek a research contract/grant with a faculty member/center at a leading institution/consortium/lab

Typical Problems:

- Prof can't put a big team on a problem overnight… progress too slow for solving pressing problems; only for blue-sky (pre-competitive) work
- Hard to clarify IP concerns, establish trust with proprietary data, etc. Hard to get company people to "buy in," exert effort on someone else's "pet project"; takes work to generate "realistic" but not actual data.
- Takes company time to guide/do course corrections on work
- Sometimes successful if DRIVEN by an executive loyal to the university, or if the university has a huge reputation/independent research arm/major research center
- **(A successful example will be discussed later)**

How Can a Company Access the Latest in EC Technology?

Plan II: Seek out a consulting company with a good reputation for solving the kind of problems you face

Typical Problems:

- Want an established company with deep pockets, reducing your risk—that's harder to find, and more expensive
- Must still use company resources in working with the consultants
- (Most reliable if it's a one-of-a-kind problem for your company and you must have and can afford the solution)

Questions?

How Can a Company Access the Latest in EC Technology?

Plan III: Find a solid company that develops/sells commercial software; license the software; get your engineers trained

Typical Problems:

- Huge investment of company time as well as $
- Only worthwhile if your people will do a LOT of this same activity
- Many vendors offer similar products; each is the best
- Your upper management may have different decision criteria than yours as an engineer/engineering manager

Questions?

How Can a Company Access the Latest in EC Technology?

- Plan IV: Track field & develop software in-house

Typical Problems:

- Much larger investment of company time as well as $
- The Common Wisdom: only worthwhile if it is *CLOSE TO YOUR CORE BUSINESS/gives you a strong competitive edge*
- Your upper management may have different decision criteria than yours as an engineer/engineering manager
- Excellent EC people disappear every year into Wall Street, Google, NetFlix, Amazon, etc. – EC is VERY GOOD at complicated data mining, for example, when combined with classical ML techniques

Academic Viewpoint: How to Introduce Your EC Technology to Industry?

- Must figure out best strategy for YOUR situation
- Must decide whether to stay INSIDE the university or go outside
- Must decide who to team up with
- Must figure out WHERE the money will come from—without source of cash, ideas are great to publish, but you don't get paid!

Ways to Get Your Search/Optimization Algorithms Used in Industry

- Plan -1: Try to patent your algorithm; maybe someone will want to license it
- Plan 0: Publish NOW, wait, *maybe* get famous, be respected, then later, *maybe* rich, as a consultant
- Plan A: Seek Contract Support from Industry through the University
- Plan B: Start an Industrial Consortium at the U
- Plan C: Start a Consulting Company
- Plan D: Start a Company to Develop/Sell Commercial Software

Plan -1: Patent and Try to License

Advantages:

- If done through university, they will pay patent costs (maybe $20-50K), but still give you a substantial part of any royalties… likely...
- No need to define YOUR IP vs. University's IP
- University's attorneys will handle any licensing negotiations, paperwork
- If university doesn't want to patent, they will likely give you rights to do so yourself, but you then pay, handle all
- (Can instead license software as trade secret…)

Plan -1: Patent and Try to License

Drawbacks:

- Others may take your ideas, run with them:
  - Patents must disclose methods/algorithms, make it possible to reproduce
  - Sometimes easy to re-implement, not make evident that they are using YOUR algorithm, or actually not use it
  - The better the algorithm (e.g., fewer parameters), the harder it is to detect infringement
  - Any legal costs to enforce patent are paid by YOU… any big corporation can run you out of money
- Maybe no one wants to license your technology; you or university is out the patent costs
- *(My advice: don't patent your algorithms—publish or keep as trade secret)*

Questions?

I am not an attorney or financial advisor. Consult yours!

Plan 0: Publish and Wait

Advantages:

- No "up-front" investment needed
- No need to define YOUR IP vs. University's IP
- Lots of citations, publicity, enhance your professional stature, may yield raises, promotion, job opportunities
- If work becomes iconic (like NSGA-II, for example), will have many companies wanting your help, paying you as a consultant or sponsoring your research

Plan 0: Publish and Wait

Drawbacks:

- Others may take your ideas, run with them:
  - You might get "passing credit" for your idea, but the implementer/marketer will get the money
  - You have no U.S. patent rights once you have published, unless you filed a provisional application for patent on/before publication date. *(NOTE: I am not an attorney. Consult your attorney for definitive information on this issue.)*
- Someone else can't patent your published algorithm, but might build from it to new patentable algorithms

Questions?

Plan A: Seek Contract Support from Industry through the University

Advantages:

- No "up-front" investment needed
- No need to define YOUR IP vs. University's IP
- Can employ "cheap" student/postdoc labor
- Through contract terms, you can benefit financially from payments to you or university without risking your own money

Plan A: Seek Contract Support from Industry through the University

Drawbacks:

- University "pace" doesn't match industry demands: they want a solution in a few months; it may take you that long to find/hire a student to start on the project
- Conditions likely needed on release of proprietary data— a problem for thesis/other publications
- Conflict-of-interest problems with supervising of students if under contract to industry, especially if YOU will get any consulting money from the sponsor
- Many companies are not comfortable obtaining IP from universities… they want someone to call for rapid response in case of problems, not a professor at a university.

Questions?

Plan A: Seek Contract Support from Industry through the University

"Secret Weapon":

- Unless you are at MIT, Stanford, Georgia Tech, etc., industry PROBABLY won't be coming to YOU looking for solutions
- Your **former students** may have high confidence in you, and as they advance into management, may be willing to contract work to you through the university
- You must still consider carefully whether you can fulfill the needs of the company, given the realities of the university research environment

Plan B: Start an Industrial Consortium at University

Advantages

- Can provide for long-term support for continuing development of a technology
- Allows companies to share costs of developing technology that is “pre-competitive” – without “collusion”
- Tends to support longer-term projects, with more university-appropriate time constants

Plan B: Start an Industrial Consortium at University

Disadvantages

- No longer as popular with companies as it earlier was – still possible at MIT, Stanford, Georgia Tech; less possible at smaller places
- Typically requires a large, already-collaborating faculty team as its core—perhaps with a specialized hardware capability
- Requires "face time" with industry reps—regular meetings, glossy reports, etc.—"management"

Plan C: Start a Consulting Company

Advantages:

- Various legal structures available, for example:
  - Sole proprietorship
  - Partnership (beware…)
  - LLC
  - Sub-S Corporation
  - C Corporation
- No need to have a fancy GUI… YOU will run the software
- YOU can tune the software, instead of the industry engineer
- University is not involved in negotiations, making them quicker and easier

Plan C: Start a Consulting Company

Advantages:

- You can (pay for and) use the same analysis/simulation software that your customers already use and trust, making it easier to get them to trust your answers
- You MAY have some control over cash flow. When you have no work, you don't pay any employees, as long as:
  - You stay **small**; YOU are doing the grunt work & marketing
  - You use "pick-up" labor, hiring ex-students, etc., when you need them, but BEWARE COI with current students
- (Of course, this makes your company less attractive)

Plan C: Start a Consulting Company

Disadvantages:

- YOU need to do all the legal work, or pay for it
- Companies may regard you as even less "stable" than a university—you have no "mass." Upper management often cares, even if engineers don't
- You don't get the help of the university in legal work, negotiations, use of university facilities, use of university-owned IP; *especially, not university-licensed commercial software!*
- YOU must resolve what IP belongs to YOU, what belongs to your university vis-à-vis the earlier/current employment relationship/contract

Plan C: Start a Consulting Company

Disadvantages (cont.):

- It's ***complicated*** to employ your current students in the consulting business… lots of conflict of interest issues. The university will be on THEIR side, protecting them from any possible abuse, regardless of how unlikely it may seem and all of your good intentions
- Customers may be WARY of your solution methods that don't give the same answer every time you run the problem, EVEN THOUGH it's THEIR tools (FEA, CFD, etc.) that tell them how the solution performs. *They translate diversity into unreliability!*
- YOU may not have the problem-related expertise of the engineer in industry to represent the problem optimally or to develop problem-specific operators, tuning, etc.

Questions?

Plan D: Start a Company to Develop/ Sell Commercial Software

Advantages:

- You make the decisions
- You keep the profits (oops, later, when there ARE profits…)
- You can decide what forms/terms of financing you are willing to accept
- You decide how to market/sell your products
- You can hire people to do what YOU don't want to do (although you must PAY them, and you need GOOD people)
- EVENTUALLY, return customers buying the software make it easier to make a profit than in a consulting business, where every contract may require new effort

Plan D: Start a Company to Develop/ Sell Commercial Software

Disadvantages:

- Must survey the market and determine what NICHE your product will fill—Are there really customers who will pay what you need for what YOU want to sell? What is the competition?
- Must pay the bills while developing a product that is ready for commercial sale and pay for marketing to grow sales
- YOU will likely work for years without a salary and putting your own money INTO the business. Can you keep your U job and take summers/ consulting time/leave time to run company? Then you may not NEED a salary from the company. Is it enough time for you in the company?
- Someone must "run" the business – work with lawyers, accountants, hire/fire, be CEO – is that what YOU want to do?
- YOU must raise working capital in order for the business to hire people, pay bills, raise funds, market, grow

Plan D: Start a Company to Develop/ Sell Commercial Software

Disadvantages (continued):

- Must be responsive to customer needs for product changes/improvements/embellishments, or you will have UNHAPPY customers, a death knell
- May need to provide many features that are not related to your "core IP" – but, "no feature, no customer"
- May need to track the competitors and "match" their product features in order to "get in the door"
- Hiring knowledgeable sales/support people is not easy, and may not be inexpensive

Questions, followed by 10-Minute Break

IEEE WCCI 2016
Vancouver Canada

Example of Plans C, D: Formation of Red Cedar Technology, Inc.

DISCLAIMER: I have not worked at RCT for 6 years. I no longer have ANY financial interest in RCT. It is now wholly owned by Siemens PLM. I have no interest in marketing RCT's current products, with which I have only limited familiarity. There may now be products on the market that are vastly superior to RCT's products! I no longer track this commercial sector.

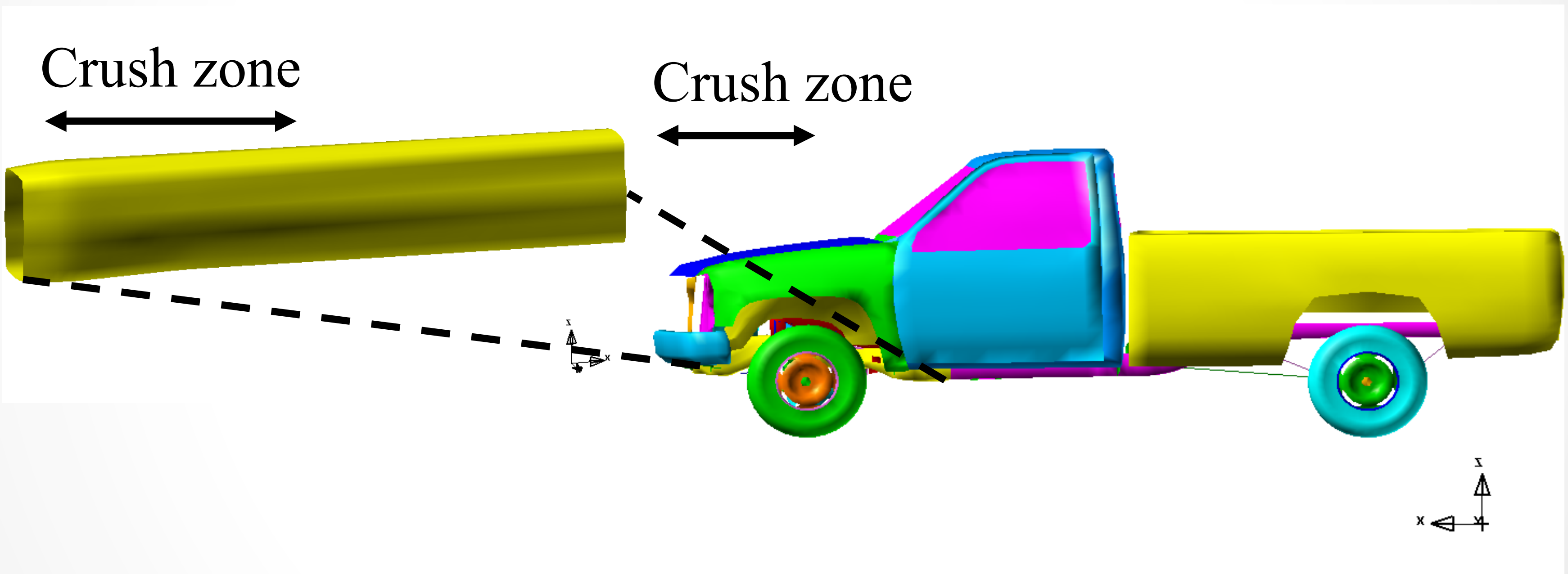
Example of Plans C, D: Steps to Formation of Company

- 1970 – I began doctoral research: first use of a genetic algorithm (not yet named that) on a real-world parameterization problem, in modeling of bacterial growth. Took ~1 year running on an IBM 1800 for one GA run
- 1973-87 Supervised 2 Ph.D. students working with GA's, including "A-Lifey" paper in 2nd ICGA, 1987
- 1993-94 – co-founded Genetic Algorithms Research and Applications Group (GARAGe) at MSU. Released (free) GALOPPS software for parallel genetic algorithms
- 1994-95 – Ron Averill (M.E., structures) and Goodman supervised students applying GALOPPS to automotive structural problems

Steps to Formation of Company (cont.)

- 1998-99 – Got a GM (Mid-Lux Engineering) contract to redesign lower compartment rail to remove weight, improve crashworthiness for a new line of cars
- Opportunity for radical redesign was opened up by ability to use *hydroforming* to manufacture rails – previous car rails were sheet steel folded into box, welded, then punched, although some truck rails had been hydroformed
- Notice: We worked with Mid-Lux Engineering in Flint, NOT optimization folks at GM Tech Center in Warren, who were still "classical" optimizers, **not interested in EC.** Problem we solved was just not solvable with their methods

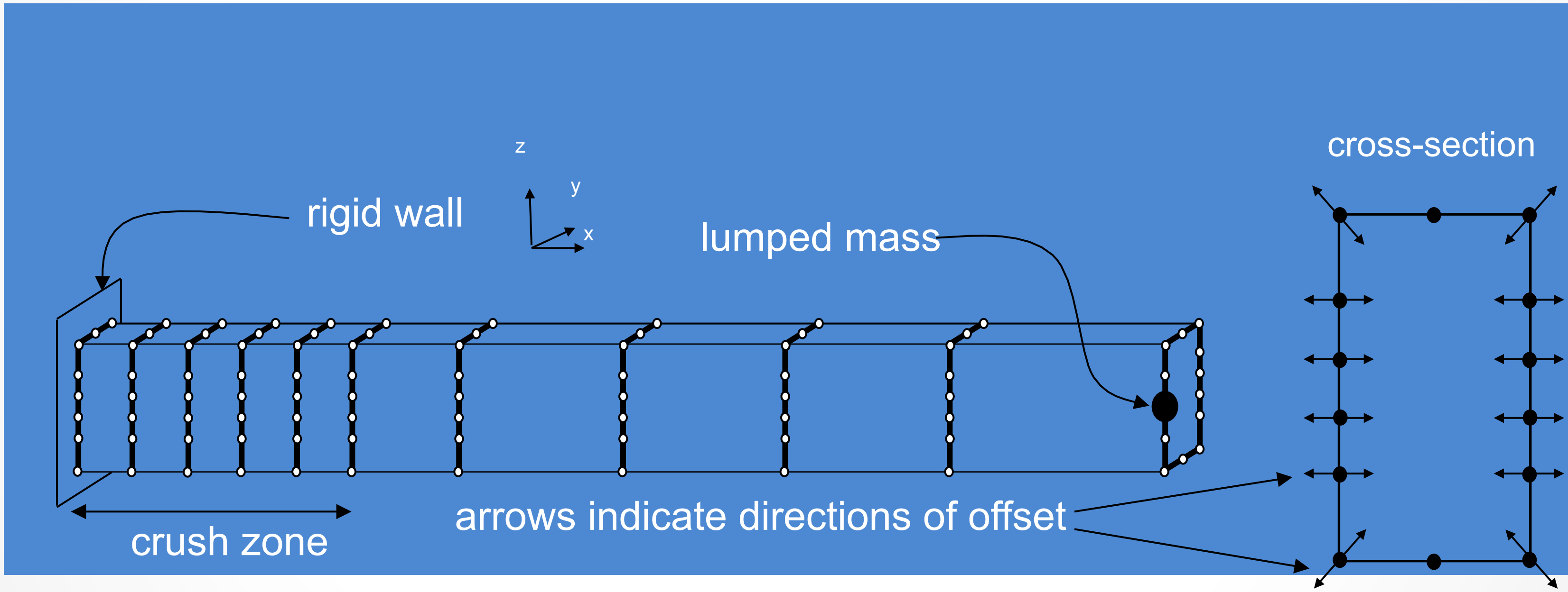
Evolutionary Design Example: Hydroformed Lower Rail



Shape Design Variables

67 design (independent) variables:

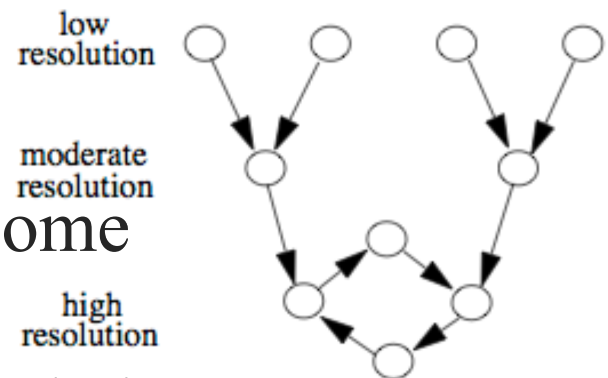
- 66 control points and one gauge thickness



Solving a Hard Problem – 1998-Style

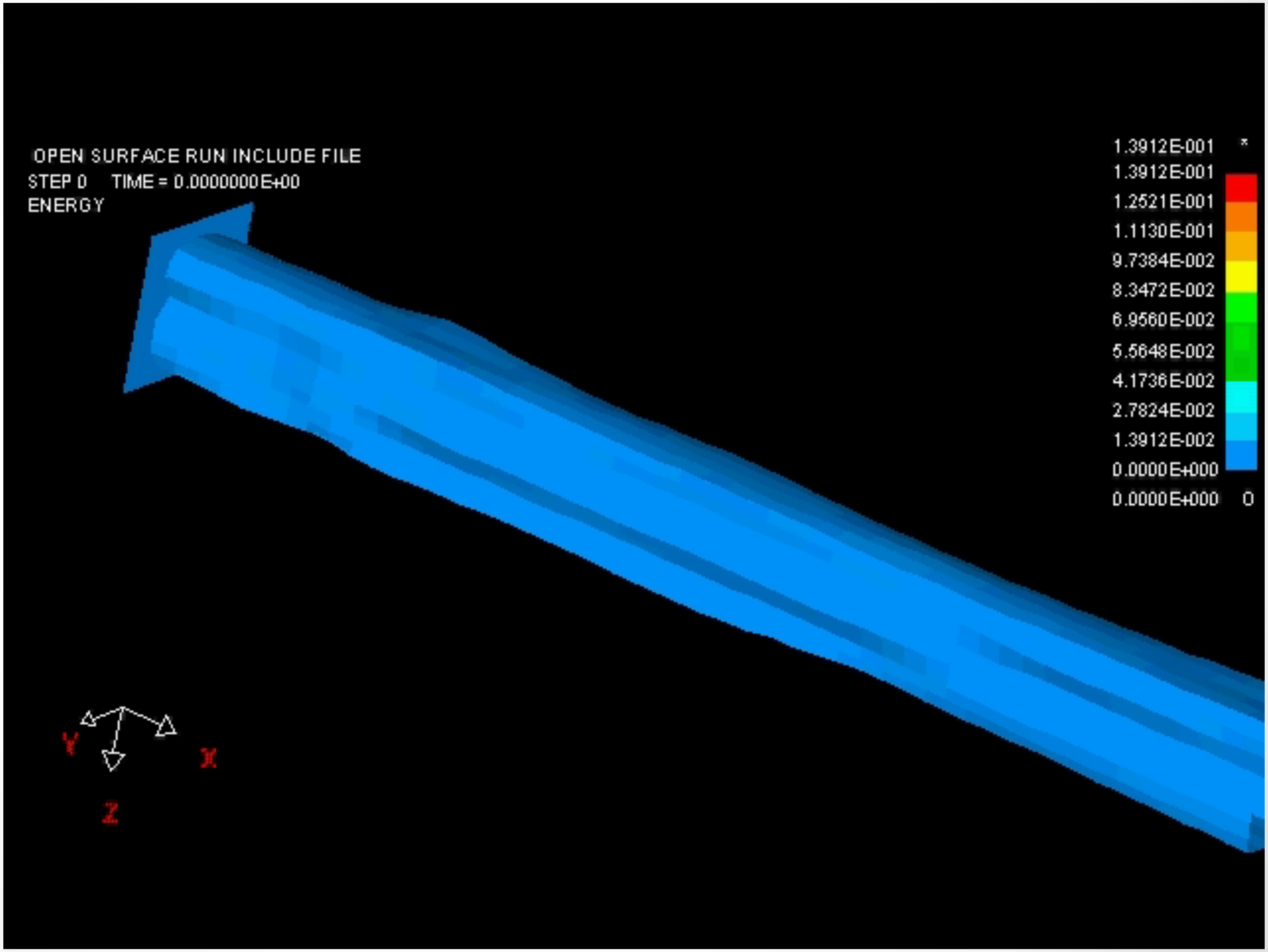
Used a 20-node PC cluster, with each node running GALOPPS with a different problem representation, using our "Injection Island Architecture":

- Some were coarse FEA resolution, some moderate, some high
- Some were stochastic, some deterministic
- At bottom level, simply island parallelism



Sample iiGA Architecture, 10 nodes

An Initial Optimized Design



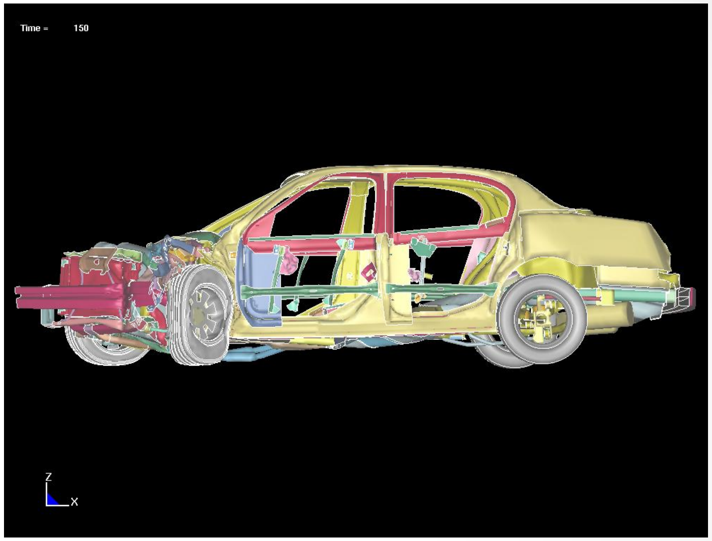
IEEE WCCI 2016
Vancouver Canada

Evolved Lower Rail Benefits

(relative to their preliminary, human-designed rail)

- Peak force reduction by 30%
- Energy absorption increased by 100%
- Weight reduction by 20%
- Overall crash response would have earned a FIVE STAR rating if car had been built

Crash Test Validation



IEEE WCCI 2016
Vancouver Canada

So Far, So Good…

To THIS point, project is an example of Industry's Plan I, Academic's Plan A:

- GM contracted with university, MSU put grad student on the problem, under Averill and Goodman
- BUT unusual in being closer to competitive technology than most university contracts

Result: When project succeeded, company wanted to get the university OUT OF THE LOOP!

Steps to Formation of Company (cont.)

1. GM Mid-Lux Engineering management said "Form a company—we'll be your customer"—1999
2. We formed ACDA in December, 1999 – Averill, Goodman, 2 grad students; **CONSULTING business**
3. Obtained license from MSU (small shareholder) to use some GALOPPS technology developed by us earlier
4. One grad student was bought out after ~3 months
5. GM backers were reorganized out of existence after a couple of years; their business went away; the new car line was dropped; we struggled to find new consulting customers
6. Second student (then Ph.D.) stayed through 2002, then left for more financial stability

Steps to Formation of Company (cont.)

7. Company name changed, went from consulting-only business into software product development; but consulting still had to pay most of the bills
8. Want recognizable name, easy to distinguish from the competition's. We chose "Red Cedar Technology" for the river running through MSU, keeping the loose tie…
9. Enter the "Business Plan." You can't raise external funds without a solid business plan—get ready to spend a lot of hours preparing it, perhaps including help from a professional to get *pro forma* financial statements in a GAAP format

Questions?

Steps In Building of Company (cont.)

7. Developed a software product called HEEDS, with proprietary (**parameterless!**) search algorithms called SHERPA and MO-SHERPA
    - Supported MDO (multi-disciplinary design optim.)
    - KEY: RAPID progress to good designs with ~200-500 evals; ***don't care about global opt of 10^{-17} at 50,000 evals!***
9. Also had to implement LOTS of "standard" search algorithms because customers were familiar with them from the software of competitors! Almost NO HEEDS customers ever USED these "standard" methods, but ***marketing*** needs FORCED us to include them
10. Senior programmer (Ph.D., M.E.) and Goodman wrote the "guts", another senior person (and shareholder) and assistants wrote OS-independent Smalltalk GUI

Steps in Building of Company (cont.)

10. **First product release in 2003**
11. We were NOT the "first in." The competition had already "spoiled" the market… customers said "We tried optimization, but it didn't work for our problems" (several earlier commercial products)
12. Making a sale meant showing the customer that it could work for THEIR "blue" brackets, since in many cases, other tools had been tried and failed. Often took 6 months
13. NOT a "shrink-wrap" product. Customer base is LIMITED: Potential customers must already be comfortable using computer-based analysis/simulation tools, or we have nothing to offer them. So pricing must yield good return on a limited market, so high value to customer!

Issues in Early Days

- Early business model: CONTINUE to use consulting income to support product development and marketing efforts
- Software licensed mostly on annual renewal basis; a few customers wanted paid-up licenses and annual maintenance fees, although we supported both models
- Had to expand cautiously—if take on a HUGE client, could we provide adequate level of support? Can't grow too quickly (but would have been happy to grow faster than we DID!!!)
- Given limited rate of growth; **cash flow is everything**!!!
- Much of labor was by founders and stockholders, with little compensation. I worked about ~800 hours/year for many years with an occasional $1,000 check when a customer paid up
- Had to deal with l-a-t-e payments, especially from automotive. We were a "supplier," and they are not kind to suppliers

Issues in Early Days

- Founders often had to give the company short-term loans to meet payroll, enable a key hire, etc. Short-term sometimes wasn't…
- Needed to develop specialized user interfaces to commonly used tools (FEM/FEA/CFD, even Excel)
- Had to develop lower-cost (less effective) methods to match competitors' low-ball entries, handle easy problems
- To keep the software improving, needed external capital – but beware of VC investment…

Venture Capital

- Can provide funding to enable rapid growth of company, in exchange for equity position
- BUT beware: VC investors often want:
  - Tight timeframe to positive cash flow
  - Very large return on investment
    - MOST VC investments fail, so when they succeed, must make up for all the failing ones, earn large ROI
  - Large equity position—your need for money gives them a lot of power
  - Strong control over an "exit event" if company goes well—either sale of company or an IPO, for example
  - Increasing level of management and board of directors control: if don't appoint their own CEO at beginning, at some point, it's likely to be triggered. Then it's not "your" company.

Issues in Early Days (cont.)

- Although we had many meetings with VC firms, we avoided taking what they offered. Have never regretted that…
- Solicited funds from early "angel" investors… Fortunately, they worked closely enough with us to have confidence in the company trajectory, and funded multiple rounds as company began to grow. Each round was smaller than what we would have gotten from VC's, but also came without the time constraints, management control
- Had to visit banks for business loans; investors and founders had to guarantee them

Success Factors

- Grew by hiring former students as consulting engineers, trainers, testers, benchmarkers, etc. (You already know what you're getting!) Most have stayed with the company. Much more difficult to hire *current* students
- Hired senior sales people with CAD/CAM sales experience; it's not easy: went through several before finding good match; must be willing to let people go if not effective
- Core software was engineered for staying power: enhanced continually, but never dumped; GUI completely redone for more flexibility, greater ease of maintenance
- We generally won our "bake-offs" by large margins. Being PARAMETERLESS was a HUGE advantage!!!
- Contracted with international resellers in several markets (India, China, Japan, Korea, Europe)
- 2010, after ~7 years of slowly growing sales (lots of repeat customers) – began to prep the company for sale

Getting the Company Ready for Sale

- Hired a CEO who had taken larger company public
- Averill still played a major role in management, closing sales, lots of travel
- Cash flow is always the growth-limiting factor
- Goodman, with award of $25M BEACON grant from NSF in 2010, withdrew from active role in company—did no more code development, but co-developer was still there, so work could progress smoothly
- Hiring continued in product development, consulting engineering, marketing and sales
- Organized International HEEDS USERS' Conference; users sang the praises of HEEDS to each other and attendees, including a SHERPA embedder, CD-Adapco ("StarCD")

The "Exit Event"

- 2013 – CD-Adapco's offer accepted to buy Red Cedar, operate it as a semi-independent business unit
- Improved cash flow situation, allowed expanding sales/marketing and development staff; growth spurted
- Key employees kept jobs (almost everyone)
- "Sweat equity" investments were recouped
- "Angel" investors made a handsome profit
- (Detailed terms, of course, are confidential)
- Red Cedar continued to grow and prosper!
- CD Adapco (and RCT) were bought in 2016 by Siemens PLM

Questions?

So, Let's Summarize and Discuss…

If you're in a company and have a problem you think EC might help to solve, you could:

- Contract with an EC faculty member developing the technology at a university/consortium/research center
- Contract with a consulting firm to deliver the solution to your problem
- License EC software providing the capabilities and train your people to use it to solve this and other problems
- Develop the in-house expertise and proprietary software to solve the problem yourselves

Each of these has many advantages and disadvantages

Summary and Discussion (cont.)

If you, an academic, have developed new technology you think has value to industry, you could try:

- Plan -1: Try to patent your algorithm or to license software as a trade secret; maybe someone will license it
- Plan 0: Publish NOW, wait, *maybe* get famous, be respected, then later, *maybe* rich, as a consultant
- Plan A: Seek Contract Support from Industry through the University
- Plan B: Start an Industrial Consortium at the U
- Plan C: Start a Consulting Company
- Plan D: Start a Company to Develop/Sell Commercial Software

Whatever Path You Take…

I wish you luck in solving your problems and advancing the frontiers of evolutionary computation!

EC has shown year after year that it can deal with difficulties that more traditional methods cannot, and I hope you get a chance to experience that yourselves!

Questions?