The tactical use of constraints and structure in diagnostic problem solving

Jeroen de Mast[a]

[a]Institute for Business and Industrial Statistics of the University of Amsterdam (IBIS UvA)
Plantage Muidergracht 12, 1018 TV Amsterdam, The Netherlands
j.demast@uva.nl

Reference:
J. de Mast (2011) "The tactical use of constraints and structure in diagnostic problem solving" *Omega* **39**(6) 702–709.

Abstract

This paper presents a prescriptive account of diagnostic problem solving, or diagnosis, in quality and process control. The paper identifies a general strategy, named branch-and-prune, whose manifestations can be found in disciplines such as medical diagnosis, troubleshooting of devices, and model-based diagnosis in artificial intelligence. The work aims to offer a clear conceptualization of this strategy, based on the notions of structures for the search space, and constraints to the cause's nature.

The idea is to treat the search space of candidate explanations as a tree structure, in which general and high-level causal directions are branched into more specific and detailed explanations. Constraints eliminate all but a few branches (pruning), which are explored in more detail. We enumerate eight generic structures as a basis for branching the search tree. We demonstrate that our conceptualization in terms of structures and constraints gives a rationale for generally known methods and heuristics in quality engineering and operations management. The paper contributes a unifying conceptual understanding of a class of diagnostic techniques, and it improves the strategy's operationality by offering generic structures, and a simpler and more flexible account of its working. A description of a real-life quality problem solving effort forms a tangible basis for the discussion.

Keywords: problem-solving; artificial intelligence; decision making/process; heuristics; learning.

1. Introduction

Diagnostic problem solving, or diagnosis, refers to the task of finding a causal explanation of observed and unwanted effects. It is typically preceded by problem formulation, and followed by the development of remedies. In business and industry, diagnostic problem-solving skills are seen as important competencies of operators, mechanics and engineers, and manufacturing companies invest in training and problem-solving methodology. MacDuffie [1], for example, presents an extensive study of problem solving on the shop-floor in the automotive industry. There is a substantial market for training and consulting services, where commercial firms offer standard or proprietary methodology such as the Shainin [2], Six Sigma [3], and Kepner and Tregoe [4] methods.

The developments in the practice of operations management are only partly reflected in the OR/MS literature on the topic of problem solving. The literature recognizes that problems come in a large variety, differing on such dimensions as the extent to which they are structured and the tasks or components they entail. Jonassen [5] proposes a typology consisting of eleven problem categories, such as logical puzzles, troubleshooting problems, and design problems. It has also been recognized that problem solving entails different sorts of tasks and subtasks, such as problem formulation, diagnosis, and the development of remedies [6]. In OR/MS, there is a substantial literature on problem structuring [7,8,9] and creative problem solving [10,11]. There has been relatively little attention in OR/MS for diagnostic problem solving, which is the topic of this work.

Motivated by empirical studies [12] which concluded that diagnostic skills can be improved by training and practice, this paper seeks to provide actionable knowledge for diagnosis. In the first place, people's problem solving skills tend to improve to the extent that they know more about the system under study; training in domain expertise is one direction for improving problem solving effectiveness [13,14,15]. Another approach to improving people's problem solving skills is to train them in the use of diagnostic methods. Typical quality problem solving courses teach methods such as cause-and-effect diagrams, multi-vari studies, and the concentration diagram [16]. The emphasis in this paper is on strategies. Wagner [14] found that experienced problem solvers typically follow different strategies than novices. In their search for a problem's cause, novices tend to apply a depth-first strategy, fixating too early on a single explanation, which they study in full detail. Experienced problem solvers avoid getting lost in problem detail by considering a variety of high-level hypotheses, and they use heuristics to select the one or a few most promising ones for more detailed exploration. Pople [13] reports a general finding in medical diagnosis, that experts use "working hypotheses", which define tentative directions in which to proceed. These working hypotheses narrow down the problem space, and bring focus to data gathering and inquiry, thus making a possibly complex, ill-structured and unbounded problem manageable. Boreham [17] observes a similar strategy in the diagnostic protocols of three experienced management consultants.

Rather than proposing an essentially new diagnostic strategy, this paper aims to clarify and conceptualize the strategic idea outlined above, which we call a branch-and-prune strategy. It addresses the challenge of how to narrow down the search space efficiently without eliminating high quality solution candidates. We propose that the quintessential principle is the tactical use of structure and constraints. We show that this conceptualization offers an insightful reconstruction of existing methods and heuristics for diagnosis. Moreover, we aim to improve the strategy's operationality by deriving tangible prescriptions for problem solvers in quality engineering and operations management.

Much of the theory that we develop is based on the literatures in other fields than OR/MS, notably, the literatures on medical diagnosis, troubleshooting of devices, and model-based diagnosis in artificial intelligence. Where relevant, we compare our findings with prescriptions found in the practitioner oriented literature, including such publications as referenced in the opening paragraph.

We start the account with a real life example, which provides context for the discussion, and may help the reader translate abstract ideas into tangible meaning. Next, we discuss the general process of diagnosis, and introduce the tactical role of constraints in a branch-and-prune strategy. Section 5 enumerates a number of generic structures which provide operational guidance for practitioners. The sixth section demonstrates that the tactical use of structures and constraints provides a unifying rationale for a selection of generally known methods and strategies for diagnostic problem solving. The concluding section discusses our findings and some of their ramifications.

2. Real-life example

To provide a realistic setting for the discussion, we begin our account with a reconstruction of a real problem solving case. The account's rather lengthy form reflects our conviction that such examples should not be streamlined to fit the authors' theories. Such streamlining results in sterile examples of problem solving, deprived of detail, which fail to bring across the confusing and messy affair that real-life problem solving often is. Without compromizing the account of the problem solving process, we did however remove references to product types, which would allow the identification of the manufacturer in question. We have verified the reconstruction below with the people involved in the project, as well as the documentation and minutes of meetings.

In the production of a certain electrical device that we shall refer to as Product *A*, a problem suddenly emerged. From week 29 in 2008 onwards, the stage in the production process where the products are tested resulted in electrical instabilities at some 12% of the products. Such an instability destroys the product, as well as the cable that connects the product to the power supply.

The first reaction to the problem consisted of checks of the process looking for things that are wrong. Some anomalies were found, such as dimensional variation in the connectors, color differences, and some pieces of equipment were polluted with chemicals. The follow-up actions on these observations did not result in useful clues. The operators and engineers also did a close inspection of some destroyed products, looking whether the instability's position and appearance would give away some clues as to its cause. As the problems persisted, the engineers also had some brainstorming-type of meetings. They identified the connecting cable as a plausible cause. In particular, there were speculations that the position of the cable connector in the receptacle was related to the problem.

All of these studies and the resulting actions did not improve the problem, and people did not have the feeling they were coming closer to understanding the problem's cause. After three months of unfruitful efforts, and with 1 out of 8 products still being destroyed, the company decided to hire an engineer with a lot of experience in problem-solving projects. This engineer, subsequently called the project leader, had been trained in problem-solving methodologies such as Six Sigma and the Shainin methods.

The project leader started by doing doing cue acquisition and exploratory data analysis on the available data. It struck him as salient that the problem had emerged very suddenly (see the attributes control chart (*p*-chart) in Figure 1); before week 29 it had never happened, whereas from that week on the occurrence rate was about 12% (48 out of 413 products destroyed in weeks 29–42). Further, it only affected Product *A*, and not Product *B*, which was similar in design and production process.

Following up on one of the outcomes of the brainstorming session, the project leader systematically analyzed the relation between the position of the cable connector in the receptacle on the one hand, and the instability occurrence on the other. The process engineers were quite suspicious of the cable and its connector. The project leader measured connector positions and related these results to the occurrence of instabilities. With values of 6.0–7.4 mm., the connectors of destroyed products clearly had deviating positions compared to the nondestroyed products (4.4–6.0 mm.). The project leader was not sure, however, that this would lead him to the cause of the instabilities, as he understood that this phenomenon need not be the cause of the instabilities, but could also be their consequence.

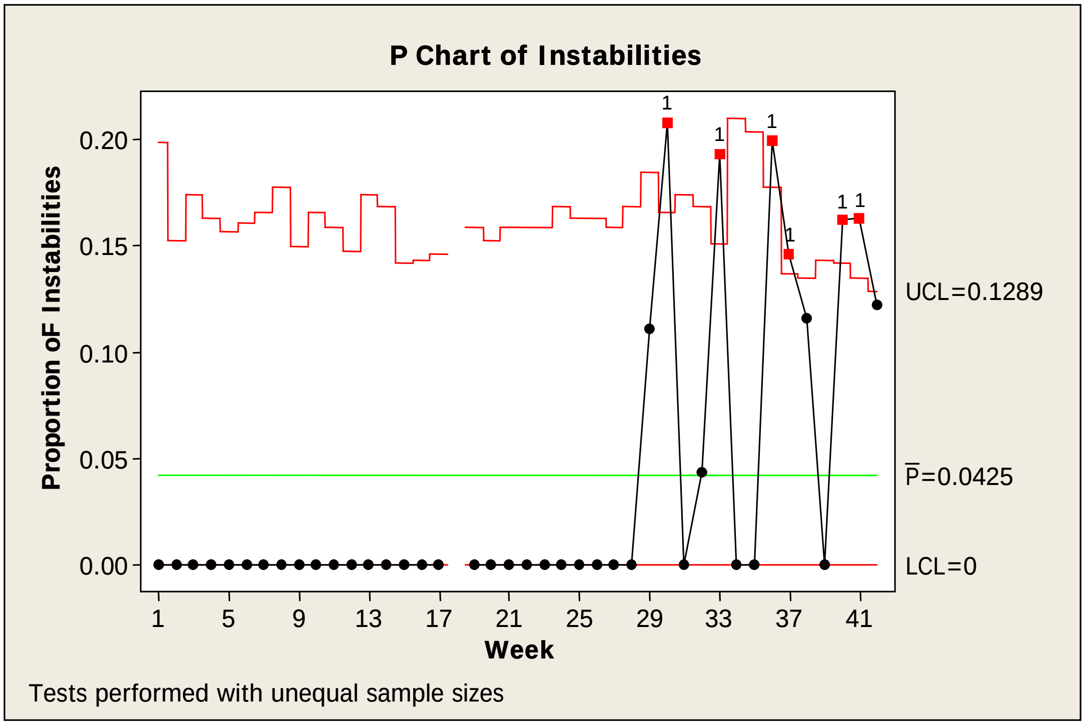

*Figure 1. Attributes control chart (*p*-chart) of the weekly proportion of products destroyed by an instability. Total volumes per week vary from 13 up to 49 (hence the nonconstant upper control limit).*

Next, the project leader started following a more systematic approach. He made a high-level map of the system under study, breaking it down into three subsystems: 1. the cable and its connector; 2. the power supply; 3. the product itself, including the receptacle and its production history up until that point. He had the power supply for Product *A* adjusted in such a way that it could be used for Product *B* as well. This allowed him to do a test that would tell him in which of the three subsystems the root cause must be found. He processed 16 *B* products with a single cable/connector; none of the 16 had an instability. A power calculation[1] showed that, if Product *B* had the same failure rate as Product *A*, the probability of observing 16 products without failure would be as low as 8.8%. The project leader concluded that the cable/connector and power supply normally used for product *A* did not give problems when used for Product *B*. Next, he processed *A* products with the same cable and the same power supply; three out of five products had an instability.

Later, it became clear that these findings were the turning point of the project, and they brought about an almost complete change of focus. The results convinced the project leader and engineers that the cause of the instabilities was in the product itself, not in the cable, the position of its connector, or the power supply, which had been the main suspects up to this point. The most relevant part of the product is the receptacle, and the project leader inventoried known issues with this sort of devices. A literature study showed that there are four known issues:

1. Dust.
2. Contamination with salts.
3. Contamination with metal particles.
4. Enclosures of air bubbles in the contact surface.

Inspired by this information, a process engineer in the team cleaned the surfaces of some of the receptacles and made a chemical analysis of the residues; they proved to contain NaCl.

[1] The project leader calculated the power from an exponential distribution, and based on the failure rate of 14% in weeks 36 through 42. A slightly more correct approach would use the binomial distribution, which would give $(1 – 0.14)^{16}$ = 8.5%.

Thinking about which aspects of the *A* and *B* products were different, the team identified the methods used in the soldering process as one of the few differences between the two. In the first weeks of the problem, the soldering process had been inspected to make sure that it was performed according to the procedures. Now, the team did a closer inspection, and the operators found that the type of soldering flux, referred to as *X*, that was used for Product *A*, but not for Product *B*, contains NaCl. *X* had been introduced recently, although it was not possible to trace when exactly. On this moment, all the pieces came together.

The last part of the project was about finding corroboration for the hypothesis that the introduction of *X* soldering flux had been responsible for the problems. A process engineer measured the conductivity of the surface of a receptacle contaminated with NaCl. The measured resistance amounts to a short circuit. The use of the *X* flux in the soldering process was discontinued as from week 43 onwards; 4 weeks later, still not a single instability had occurred. After three months of unfruitful attempts, the project leader's structured approach had led to the discovery of the root cause of the problem in about 4 weeks.

3. The diagnostic process

Evaluating the problem solving process in the example, we note that the team tried a variety of problem solving strategies, ranging from relatively cursory to methodical and penetrating. Before we study the use of structures and constraints in diagnosis, we discuss the general reasoning process in diagnosis.

Diagnosis can be typified as reasoning from behavior to its causes; confronted with some form of unwanted behavior or malfunctioning, the problem solver must infer the causal chain of events or the structural aberration that produces it [14,18]. We discuss a number of basic elements of the diagnostic process [cf. 14, 15, 19].

1 Building or expanding domain and fault models

To hypothesize about potential causal explanations for a malfunction, the problem solver needs mental models about the system under study, which together are called a domain model [20]. The domain model typically includes a structural or anatomical model (that is, a decomposition of the physical system into subsystems and components and their linkages), perhaps in conjunction with a functional or causal model (that is, a model specifying how the subsystems and components interact causally or functionally). It may also include a representation of the system's history and development over time.

In addition to the domain model, the problem solver may possess or develop fault models, which capture knowledge derived from earlier malfunctions, and thus represent the accumulated experience with the system under study or similar ones. They could include taxonomical classifications or fault dictionaries of known malfunction types. They could also have the form of decision trees or other rule-based systems, which associate symptoms with faults that have caused them in the past, or which prescribe a sequence of tests to guide diagnosis. Among the fault models typically are collections of representations of earlier problematic episodes. As a matter of fact, an important part of being an expert in a domain consists of having a large collection of such earlier episodes in memory. In case-based reasoning [21], libraries of past episodes are purposely built to aid future problem solving efforts.

Domain and fault models may be stored on paper or other storage facilities, or they may be kept in the problem solver's memory. Much of the literature in artificial intelligence concerns diagnosis by expert systems, to which the domain and fault models are given in an explicated, consistent, and fairly complete form. The focus in this paper is on human problem solving, where typically, at the start of the diagnostic process, only fragments of domain and fault models are available to the problem solver, and in a variety of forms. Substantial parts are built as insight advances during and as part of the problem solving process. Hence, building and expanding the domain and fault models is a recurring element in the diagnostic process. The project leader in the example expanded his domain models on many occasions, as when he learned that the soldering process is the main difference in the production of *A* and *B*, that there is such a thing as *X* soldering flux, that *X* flux contains NaCl, and that *X* is used for Product *A* but not for *B*. His literature search, resulting in a list of four known issues with the type of receptacles he was dealing with, was a deliberate effort to augment his fault models.

2 Making observations

The diagnostic process is typically instigated by observations symptomatic of a malfunctioning. Also during the diagnostic process the problem solver makes observations, sometimes with a well-defined aim dictated by the diagnostic process, sometimes casually and without predefined plan. Some observations are recognized as symptoms of the problem under study; for other observations it may not be clear whether they are related to the problem. Observations may be quantitative data, but also less structured sources of information such as impressions and anecdotes.

3 Generation of hypotheses

A recurrent activity in the search for the true causes is the generation of possible explanations, which is done on the basis of knowledge about what sort of things there are, how they interact and developed over time, and what sort of malfunctions have occurred in the past; in other words, on the basis of domain and fault models. Hypotheses range from detailed and specific causal explanations to general directions of causes.

Hypotheses may be generated by deep reasoning, also called model-based diagnosis or diagnosis from first principles [15,22,23]; it has been studied extensively in artificial intelligence. Deep reasoning refers to its being based on an understanding of the physical structure and working of the malfunctioning system as embodied by the domain models. Candidate causes are identified by abductive reasoning from symptoms in conjunction with the domain model: “Given the structural and functional model, the symptoms would be a logical consequence of *X*, so *X* may be the cause.”

Deep reasoning is often contrasted to shallow reasoning, also called symptomatic search, which is based on experiential association of symptoms to potential causes [19,22,23]. Symptoms are not colligated with the domain models, but with the fault models, thus seeking to relate the current malfunction to earlier experiences with the system. The symptoms may match a fault category in a taxonomy or fault dictionary. Or the symptoms may remind the problem solver of an earlier, similar episode. Or the problem solver may use a search engine to find earlier experience with similar symptoms. For example, the author of this paper solves many computer problems without deep understanding of the problem, but shallowly by typing the error message verbatim in a search engine on the internet, and following the pieces of advice that the subsequent search produces. Essentially, shallow reasoning yields findings of the form “*X* may be the cause, since the symptoms are known to have been caused by *X* before”.

The example project shows a combination of shallow and deep reasoning, and perhaps even altogether different forms of reasoning, as do most real diagnoses. Especially the early attempts, consisting of cue acquisition, autopsies, and checking the usual suspects, are manifestations of shallow reasoning. Such approaches are often effective in every-day routine problem solving, their rationale being the “common things are common” heuristic [13]. But shallow reasoning is rather ineffective for novel problems [22], and the early attempts did not result in useful leads. When the amount of accumulated information and clues started to become unmanageable and confusing, the project leader realized that he needed a more penetrating approach. Note, however, that also later in the project instances of diagnosis by shallow reasoning abound, such as the literature search resulting in four known issues with receptacles.

4 Testing hypotheses

Hypotheses are tested in hypothetico-deductive style: the problem solver derives logical implications of possible explanations, and compares these to (old or new) observations, rejecting hypotheses whose implications are contradicted by observations. Alternatively, evidence may be so strong that a hypothesis is adopted as the definitive explanation, thus ending the diagnostic search. The project leader tested, for instance, whether the position of the connector is related to the instabilities (which did not result in a decisive refutation or confirmation), and whether NaCl in *X* is the root cause of the problem.

5 Search tactics
If 1, 2, 3 and 4 were the only types of activities, diagnosis would quickly degenerate into a random trial-and-error search. Efficiency of the diagnostic search is determined by search tactics, which sequence the activities described under 1, 2, 3 and 4.

4. A branch-and-prune strategy based on structure and constraints

Within the framework just expounded, we aim to clarify a class of search tactics that we name *branch-and-prune*. The search space in a diagnostic search is the field of all possible causal chains that could explain the malfunction under study. Of course, this space is often infinite and it is highly unlikely that the problem solver has an exhaustive description. A branch-and-prune strategy treats the search space as a tree structure, in which general and broad causal directions are branched into more specific and detailed causal explanations. Instead of elaborating the search tree in its entirety, the strategy prescribes an iterative process in which high-level branches are pruned, and only the retained directions are elaborated into more detailed branches. The search space is progressively narrowed down until a sufficiently specific explanation is found. This strategy resembles the branch-and-bound meta-heuristic in mathematical programming [24].

Diagnostic efficiency can be compromised by excessive divergence of the search tree, as when one starts with a brainstorm session resulting in a large multitude of possible explanations, all of which need testing. Another danger is a lack of divergence, as when one fixates prematurely on a single explanation. Branch-and-prune tries to balance divergence and convergence by working iteratively on varying levels of specificity, limiting the development of domain and fault models, as well as hypothesis generation and testing, as much as possible to the relevant parts of the search tree. The basic idea is often promoted for its efficiency, for instance, in artificial intelligence, where it is known as hierarchical model-based diagnosis [23]. There is empirical evidence [13,17] that experienced consultants and physicians work according to this strategy, often with the suggestion that this strategy is efficient or prevents diagnostic errors (e.g., "… the failure to employ a top-down refinement strategy (…) may be a critical factor in the failure of subjects to diagnose the cause of the problem." [25]). Later in this paper we discuss variations on this strategy proposed in the practitioners' literature.

In the example, the project leader branched the search tree into three general classes of causes, related to the three components (cable, power supply, product itself) of the physical system. His experiment allowed him to discard causes related to the power supply and the cable from further consideration; homing in on the product itself, the project leader invested time in learning more about its receptacle, and found a standard taxonomy of four known issues for this sort of devices. Likewise, upon identifying the soldering process as a relevant process step, the team studied this part of the process in more detail, and discovered the soldering flux containing NaCl.

In this paper we develop a conceptualization of this strategy based on the notion of constraints. We use this term in a more general sense than in mathematical programming, as referring to hypotheses that rule out regions of the search space. In the example project, the project leader identified these constraints to the cause's nature:

- The root cause is something that changed in week 28.
- The root cause is something that differs for Products *A* and *B*.
- The root cause is not in the cable/connector, nor in the power supply, but in the product itself (and thus upstream in the production process).
- The root cause is probably in the soldering process.

- The root cause must be able to leave an NaCl residue.

The combination of these constraints left only a limited subset in the search space, in which the root cause could be identified by an exhaustive search.

Constraints presuppose that the problem solver has in mind a structure for the search space. A structure is a collection of relations among elements, and these relations define classes. A constraint is the exclusion of certain classes from further investigation. There are limitless relations, and therefore, many ways to structure the search space. In some cases, a structure is defined by the problem solver; in the example, the project leader derived a structure from the physical system's three components. Deliberate testing revealed the constraint: the relevant class are causes associated to the product itself, eliminating the cable and the power supply subsystems from further investigation. In other cases, a structure is suggested by observed behavior. For example, the observation that the weekly fraction of destroyed products rose from zero to about 12% in week 29 suggested a temporal structure, consisting of the classes "events before week 28", "events in weeks 28–29", and "events after week 29". The same observations revealed the constraint that the relevant causes are in the "events in week 28–29" class. Structure may also be provided by fault models; for example, the organization of a taxonomy providing categories of malfunctions is a structure.

In creative problem solving constraints are typically seen as a bad thing [10], which prevent problem solvers from thinking "out of the box" (although there is evidence that in art, which could be conceived as a highly open and ill-structured form of problem solving, self-imposed constraints force the artist to leave well-trodden paths, and thus stimulate novelty [26]). We pose that in diagnostic problem solving, constraints can be a good (that is, tactically useful) thing, which make diagnosis more efficient.

5. Generic structures for diagnostic search

A more precise account of the role of constraints in diagnosis may help practitioners in exploiting constraints more consciously and thoughtfully. In addition, we offer eight generic structures that problem solvers may exploit for identifying constraints and focusing their diagnostic effort. This enumeration is based on a large number of problem solving cases, but it is not complete.

1 Physical structure

For products and technical systems, a structure can be defined in terms of a breakdown into physical subsystems and elements, such as modules, components, and parts. For production processes, the breakdown can be in terms of machines, workstations, cavities in a machine, position on a conveyor belt, and further into the details of the physical installations present. Consider a person who connects his or her computer to a projector, but the projector indicates that it does not detect an input signal. The system could be discerned into the computer, the projector, and the cable connecting the two. An efficient strategy is to connect another computer to the projector; if the projector detects the signal, the problem is in the original computer. If the problem persists, the next step could be to try another cable, thus establishing whether the problem is in the cable or in the projector. As a result of this strategy it is likely that two out of three subsystems can be completely discarded from the diagnostic effort.

2 Spatial structure

Spatial structures are the relations among elements in space, such as the lay-out of a shop-floor (and constraints could exclude sections of the lay-out from further investigation). Also, the surface planes of a product constitute a spatial structure (constraints possibly isolating a particular spot on a surface plane for further study).

In a famous episode in the discovery of the transmission mechanism of cholera, a doctor named John Snow pinpointed the source of a local outbreak in London by observing that the casualties clustered geographically in a certain area in London [27], and he zoomed in on this area. Aware of the possible role of the drinking water system in cholera epidemics, Snow identified a certain water pump in the cluster's epicenter as potentially instrumental in the outbreak.

3 Sequential structure

A production process consisting of consecutive steps is a sequential structure. By splitting the process in halves, and observing whether the aberrant behavior is already manifest halfway, one may establish whether the cause is before or after that point. The technique is routinely used in the debugging of computer programs. Given a single module consisting of a large number of lines of code, and confronted with the problem that the program crashes when executed, the programmer may add a 'print' command somewhere halfway in the program, and thus establish whether the program crashes before or after that point. By repeating the procedure, he or she zooms in on the offending line.

4 Product families

A breakdown of products into types and families gives a structure for the search tree. In this paper's example, the project leader noted two types of products (*A* and *B*), and observing that the problem occurred only for products of type *A* identified the constraint that "the root cause is something that differs for products *A* and *B*".

5 Diachronic structure

Diachronic refers to the development of phenomena through time. Some examples of constraints defined in terms of diachronic structures are:
- The root cause occurred in (or before) week 28.
- The root cause is something that aggravates over time.
- The root cause varies from batch-to-batch (or day-to-day, shift-to-shift, etc) and is constant within batches (days, shifts).

6 Organizational structure

Organizational structure refers to the organization of activities in locations, sites, production streams, shifts, production lines, and workflows. Suppose the electrical instabilities had occurred in one production site, but not all other sites, one would have the constraint that the cause is something specific for the problematic site.

7 Functional and causal structure

Physical components may be tied to specific subfunctions. If that is the case, observing which subfunctions are delivered as normal, and which are aberrant, one may exclude certain components, and focus on others. Suppose one debugs a computer program consisting of a multitude of subroutines, and suppose that these subroutines are associated to functions and subfunctions of the program. Observing or testing which of the program's functions work normal, and which are invalid, one infers which subroutines to focus on.

In similar vein, one may try to reconstruct the chain of causation producing observed symptoms, hoping that upstream in the chain one identifies a cause that is related to particular physical components, but not to others [22]. In the example, the discovery of a salt residue on the receptacle's surface resulted, as a working hypothesis, in the reconstructed causal chain *salt residue → short circuit → electrical instability*. This constrains the search for root causes to a limited number of process parts by excluding everything that cannot plausibly be expected to leave a salt residue.

The abovementioned doctor Snow, when searching for the transmission mechanism of cholera, noted that the disease affects the alimentary canal first, not the lungs or blood, and inferred that the disease agent is swallowed in, not breathed in (reconstructing a plausible chain of causation from observed symptoms). This ruled out scenarios involving air-borne disease agents, and focused the search on substances that enter the victim's body orally, such as the drinking water (see De Mast and Kemper [27], and references therein).

8 Structure suggested by taxonomies

Taxonomies of known malfunction types are often organized as a hierarchy or decision tree, thus providing a structure for identifying constraints. The taxonomy prescribes a series of tests or observations aimed at ruling out branches of the search tree. Lucas [20] gives an example from medical diagnosis, where disorder taxonomies prescribe that, in diagnosing jaundice of hepatobiliary origin, a clinician should first try to determine whether the disease affects the liver cells or the biliary tract; suppose it's the liver, then the taxonomy prescribes the clinician should next establish whether the disorder is acute or chronic.

6. Methods and strategies employing structure and constraints

We discuss a number of known methods and strategies, proposing that at least part of their effectiveness is their ability to suggest constraints that direct attention towards the problem's cause, thus bringing them in the unifying framework of branch-and-prune strategies. Where appropriate, we suggest how the generic structures and a more deliberate pursuit of constraints may improve the power of these methods.

Cue acquisition

Cue acquisition is the systematic collection of symptoms, such as done by physicians in a patient work-up. In solving quality problems, it could be guided by generic questions such as Who? Where? When? What? How? How much? (known under the acronym 4W2H). Another form of cue acquisition are autopsies [16], which are close examinations, typically involving a disassembly, of malfunctioning parts or products.

One way in which cue acquisition works, is that it facilitates diagnosis by shallow reasoning. Symptoms may match with entries in a fault dictionary, or they may remind the problem solver of a previous, similar episode (or the problem solver may enter the symptoms in a search engine, and find a matching case in that way).

More topical to our discussion is that symptoms may reveal constraints, and thus focus the search. Note how the 4W2H questions probe the search space by trying different structures, such as spatial (where?), temporal (when?), and functional (what?) structure. Instead of 4W2H, the generic structures proposed in this paper may be used as a guidance in cue acquisition.

Finally, symptoms may lead to the discovery of constraints by suggesting functional or causal structure, as the discovery of an NaCl residue inspired the causal chain of *salt residue → short circuit → electrical instability*, and the ensuing constraint "The offending part must be able to leave an NaCl residue."

Contrasting

Here, we refer to the systematic comparison of the problematic to the unproblematic, as is done in pairwise comparison, where the *best-of-the-best* (BOB) products are compared to the *worst-of-the-worst* (WOW) products [2]. Kepner and Tregoe's "*is* versus *is not* analysis" [4] has the problem solver identify what distinguishes objects, behavior, locations and situations where the problem is from those where it could be but is not.

The project leader noted that the instabilities only occurred in products of type *A*, and not in *B* (applying a product family structure) and discovered the constraint that "the cause must be something that differs for products *A* and *B*." Combined with the earlier identified constraint that "the cause is in the product itself, not in the power supply or cable," and the fact (given by the project leader's domain models) that soldering is the main difference in the production processes of *A* and *B*, it narrowed down the search space to causes related to the soldering process.

Contrasting works in ways similar to cue acquisition, but by considering extremely good and extremely bad specimens, the relevant symptoms may be more pronounced or easier singled out from accidental features; Steiner, MacKay and Ramberg [2] refer to this principle as *leveraging*. Contrasting often produces constraints of a particular kind, which we name *contrasts*. They have the form "The cause must have two different states, one co-occurring with the problematic (product type, location, time period, …), and the other co-occurring with the unproblematic," for example, "the cause must be something that differs for products *A* and *B.*" This paper's generic structures provide useful dimensions for contrasting.

Exploratory data analysis

Exploratory data analysis (EDA) is the mainly graphical evaluation of data aimed at finding salient features [28]. These salient features could be recognized by the problem solver as the fingerprint of a certain phenomenon (shallow reasoning). But often, they reveal constraints. De Mast and Trip [28] describe many examples of problem solving using EDA, including a Pareto chart which helped in going from a broad problem conception ("too many defects") to a more focused notion ("too many defects from workstation no. IX"), thus exploiting an organizational structure (namely, a categorization by workstation). In another example, a histogram of eccentricity measurements showed a bimodal distribution, revealing the constraint that the main cause of eccentricity must have two clearly distinguishable states (as it turned out, there were two molds in the process, and one of them was worn out).

EDA may be applied more effectively by systematic use of the proposed generic structures as dimensions in graphs. Diachronic structure is at the basis of time series plots (as in Figure 1); spatial structure is exploited in defect concentration diagrams [16]; and product families and physical or organizational breakdowns can be the basis for group comparison techniques such as ANOVA, boxplots per group, and analysis of means.

Besides its role in identifying constraints, EDA may also facilitate the selection of interesting objects and specimens for autopsies and pairwise comparisons (as defined above).

Multi-vari studies

Especially for diagnosing problems with excessive variation, practitioners' books in quality engineering [16,29] suggest that one establish whether the dominant source varies over time, between production streams, from piece to piece, or within pieces. The multi-vari chart [16,30] is a graphical technique for this purpose. Multi-vari studies identify constraints from diachronic, organizational and physical structures.

Methods such as brainstorming and cause-and-effect diagrams, popular for diagnosis in practice, are not included in this discussion, as they do not facilitate a branch-and-prune strategy, but instead, serve other functions in diagnosis (for example, the two mentioned methods facilitate the hypothesis generation task).

In addition to methods, literature describes a number of strategies that are, in our view, manifestations of the branch-and-prune strategy described in this paper. They are typically limited to the exploitation of one or a few of the generic structures described earlier.

Half-split strategy

The half-split strategy, also named bisection [12], is a generally known heuristic with diverse applications. It is tied to sequential structure such as the sequence of steps in a process. The strategy has the problem solver establish whether a problem is already manifest halfway in the process, and subsequently exclude the first or second half. Recursively applying the heuristic, the problem solver homes in on a small part of the process.

Topographic search

A topographic search [19] exploits functional structure in conjunction with a domain model of the physical structure of the malfunctioning system. Starting from the functions that are delivered abnormally, one identifies which parts of the system have a downstream, causal link to the malfunction, and eliminates everything else. The strategy is typically applied recursively, applying it on the level of whole subsystems in the early phases, to individual components in the final stage.

Shainin's progressive search

Progressive search was popularized in industry by Shainin [31] as a strategy for variation reduction. Framed in colorful terminology, the strategy is an example of branch-and-prune in which the search tree is branched in families of variation sources [2]. Following a process of elimination the problem solver homes in on the class containing the dominant causes, guided by techniques such as the multi-vari chart, pairwise comparison (BOB vs. WOW), and component swapping. The latter exploits physical structure to establish in which of a number of components the root cause may be found.

Kepner and Tregoe's problem analysis
Problem analysis is one of the four rational processes of Kepner and Tregoe. We comment on its procedure, as given in [4].

- The method insists that the problem is described in specific terms, guided by the dimensions *what*, *where*, *when* and *extent*. These dimensions serve a similar purpose as the generic structures identified in this paper.
- The problematic situation is compared to situations where the problem could be but is not, again using the dimensions *what*, *where*, *when* and *extent*. This results in constraints of the type that we have named *contrasts*: causes to be considered must have two states, co-occurring with the problematic and unproblematic. Application of these steps to the problem of electrical instabilities might reveal that the problem is manifest in product *A*, and could be but is not in product *B*, which gives the contrast that "the cause is something that distinguishes products *A* and *B.*"
- Potential causes could now be discovered from knowledge and experience (shallow reasoning), or they are discovered by considering distinctions. In the latter case, the problem solver identifies what distinguishes the problematic *what*, *where*, *when* and *extent* from the unproblematic, thus generating hypotheses that could account for the identified contrasts. The contrast above could motivate a close study of the soldering process, because it is a distinction between products *A* and *B*.
- Next, considering what has changed in, on, around, or about these distinctions, the problem solver applies diachronic structures to focus on distinctions whose evolution over time agrees with the history of the problematic behavior. In the example, the problem solver is urged to consider what has changed in the soldering process in or around week 29.
- The process is concluded by the testing of possible causes, the establishment of the most probable one, and an attempted confirmation.

The description of the method in [4] does not describe the branching and pruning principles, nor does it mention explicitly the iterative nature of the process. Instead of a search tree that branches on several levels of increasing specificity, the procedure suggests only a single focusing step, in which several structures (inspired by the *what, where, when* and *extent* dimensions) are applied in conjunction. We think that the exposition of the method could gain in clarity by employing the concepts of recursion, and branch and prune. The method is somewhat limited by only considering constraints of the type of contrasts. Finally, this and the other strategies discussed in this section are limited in considering only a subset of the eight generic structures; Kepner and Tregoe's problem analysis, for example, does not exploit sequential structure and the associated half-split strategy explicitly.

6. Discussion and conclusions

Problem solving entails a variety of tasks, such as problem formulation, diagnosis, and the development of solutions, the latter being the subject of the literature on creative problem solving. Due to differences in purpose and process, techniques and strategies for creative problem solving [32] should be distinguished from those for diagnosis. Where the creative development of remedies and solutions thrives on divergence and lateral thinking, diagnosis, as conceptualized in this paper, is predominantly convergent, and the dominant form of reasoning is analytical rather than creative. This is reflected in the different role that constraints on the search space play; namely, constraints as an impediment to novel ideas in the search for remedies, versus constraints as a tactically useful principle in diagnosis. However, also in diagnosis there may be creative steps. The generation of hypotheses, for example, may often be an uncreative enumeration of all components in (part of) a system, or of all disorders in a taxonomy, but sometimes it involves the creative invention of novel explanations.

The branch-and-prune strategy could be summarized as follows. Especially in the early stages of diagnosis, and after the usual suspects have been checked, the advice is not to lose oneself in testing individual candidate causes, but instead, to invest one's time in proposing structures that define classes of causes in the search tree ('branch'), and interpreting observations for deciding which class to concentrate on ('prune'). In some cases, the evidence may be fairly conclusive, and the working hypothesis may be accepted with reasonable statistical confidence. In other cases one has to make do with less conclusive evidence, or one retains more than one working hypothesis. Even in the case where strong empirical evidence is available, the task of putting forth structures is an essentially speculative and fallible one, and even in such cases, diagnosis does not typically follow inflexible and infallible algorithms (as sometimes claimed in the practitioners' literature [29]). Moreover, the branch-and-prune strategy will typically be employed in conjunction with other strategies. Diagnosticians behave opportunistically, adjusting activities within a strategy and changing strategies in response to information and ideas [15,17,25].

This paper contributes a conceptualization of a class of diagnostic heuristics, framed in terms of structure and constraints for the search space. The account offers a unifying, insightful and flexible framework for understanding diagnostic methods and strategies commonly applied in quality engineering and operations management. Moreover, the framework, and especially the eight generic structures and associated tactical moves, improve the strategy's operationality.

References

[1] MacDuffie JP. The road to 'root cause': shop-floor problem-solving at three auto assembly plants. Management Science 1997; 43:479–502.

[2] Steiner SH, MacKay RJ, Ramberg JS. An overview of the Shainin System for quality improvement. Quality Engineering 2008; 20:6–19.

[3] De Koning H, De Mast J. A rational reconstruction of Six Sigma's Breakthrough Cookbook. International Journal of Quality and Reliability Management 2006; 23:766–787.

[4] Kepner CH, Tregoe BB. The New Rational Manager. Princeton: Kepner-Tregoe; 1997.

[5] Jonassen DH. Toward a design theory of problem solving. Educational Technology Research and Development 2000; 48:63–85.

[6] Smith GF. Towards a heuristic theory of problem structuring. Management Science 1988; 34:1489–1506.

[7] Volkema RJ, Evans JR. Managing the process of formulating the problem. Interfaces 1995; 25:81–87.

[8] Mingers J, Rosenhead J. Problem structuring methods in action. European Journal of Operational Research 2004; 152:530–554.

[9] Paucar-Caceres A. Mapping the changes in management science: A review of 'soft' OR/MS articles published in *Omega* (1973–2008). Omega 2010; 38:46–56.

[10] Ackoff RL, Vergara E. Creativity in problem solving and planning: a review. European Journal of Operational Research 1981; 7:1–13.

[11] MacCrimmon KR, Wagner C. Stimulating ideas through creativity software. Management Science 1994; 40:1514–1532.

[12] Morris NM, Rouse WB. Review and evaluation of empirical research in troubleshooting. Human Factors 1985; 27:503–530.

[13] Pople HE. Heuristic methods for imposing structure on ill-structured problems: The structuring of medical diagnosis. In: Szolovits P, editor. Artificial Intelligence in Medicine. Boulder, CO: Westview Press; 1982, pp. 119–190.

[14] Wagner C. Problem solving and diagnosis. Omega 1993; 21:645–656.

[15] Smith GF. Determining the cause of quality problems: Lessons from diagnostic disciplines. Quality Management Journal 1998; 5:24–41.

[16] Gryna FM. Quality improvement. In: Juran J, editor. Juran's Quality Handbook, 4th ed. New York: McGraw-Hill; 1988, pp. 22.1–22.74.

[17] Boreham NC. A model of efficiency in diagnostic problem solving: implications for the education of diagnosticians. Instructional Science 1986;15:191–211.

[18] Davis R, Shrobe H, Hamscher W, Wieckert K, Shirley M, Polit S. Diagnosis based on description of structure and function. Proceedings National Conference on Artificial Intelligence (August, 1982); 1982, pp. 137–142.

[19] Rasmussen J. Models of mental strategies in process plant diagnosis. In: Rasmussen J, Rouse WB, editors. Human Detection and Diagnosis of System Failures. New York: Plenum Press; 1981, pp. 241–258.

[20] Lucas PJF. Model-based diagnosis in medicine. Artificial Intelligence in Medicine 1997; 10:201–208.

[21] Aamodt A, Plaza E. Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications 1994; 7:39–59.

[22] Davis R, Hamscher WC. Model-based reasoning: Troubleshooting. In: Shrobe HE, editor. Exploring Artificial Intelligence. San Francisco: Morgan Kaufmann; 1988, pp. 297–346.

[23] Mozetič I. Hierarchical model-based diagnosis. International Journal of Man-Machine Studies 1991; 35:329–362.

[24] Lawler EL, Wood DE. Branch-and-bound methods: A survey. Operations Research 1966; 14:699–719.

[25] Smith PJ, Giffin, WC, Rockwell TH, Thomas, M. Modeling fault diagnosis as the activation and use of a frame system. Human Factors 1986; 28:703–716.

[26] Stokes PD. Using constraints to generate and sustain novelty. Psychology of Aesthetics Creativity and the Arts 2007; 1:107–113.

[27] De Mast J, Kemper B. Principles of exploratory data analysis in problem solving: What can we learn from a well-known case?, with discussion and rejoinder. Quality Engineering 2009; 21:366–383.

[28] De Mast J, Trip A. Exploratory data analysis in quality improvement projects. Journal of Quality Technology 2007; 39:301–311.

[29] Bhote K. World Class Quality. New York: Amacom; 1991.

[30] De Mast J, Roes KCB, Does RJMM. The multi-vari chart: A systematic approach. Quality Engineering 2001; 13:437–447.

[31] Shainin R. Strategies for technical problem solving. Quality Engineering 1993; 5:433–448.

[32] Summers I, White DE. Creativity techniques: Toward improvement of the decision process. The Academy of Management Review 1976; 1:99–107.