Future States for a Present-State Estimate, in the Contextual Perspective of In-Core Nuclear Fuel Management

E. Nissan,* A. Galperin, A. Soper,* B. Knight,*** and **J. Zhao***

(*) School of Computing and Mathematical Sciences, The University of Greenwich,
Queen Mary Court, 30 Park Row, Greenwich, London SE10 9LS, England, U.K.
E-mail: `E.Nissan@gre.ac.uk` `http://www.gre.ac.uk/~E.Nissan`
`http://www.gre.ac.uk/~E.Nissan/CmsWeb`

(**) Department of Nuclear Engineering, Ben-Gurion University, Beer-Sheva, Israel.
E-mail: `alexg@bgumail.bgu.ac.il`

Abstract

We discuss AI-based and related approaches to problem-solving for an allocation design problem, with which fuel managers at nuclear power plants are faced on a peri-annual basis: how to position fuel-units in the reactor core (whose planar section is a symmetric grid), to achieve better, and usually better performance during the operation period up to the next EOC point (end of cycle), when over a new downtime period the allocation problem will have to be solved again. Forecasting in this domain is not accurate enough to enable the preparation of robust solutions before the reactor has actually been shut down, and the depletion degree of the fuel-units, let alone their possibly damaged status, can be ascertained. Various approaches exist. Westinghouse's LPOP is based on backcalculation from a target power-distribution. In contrast, the FUELCON expert system (Galperin and Nissan, 1988) applies hyperrecursion on a heuristic ruleset to generate alternative candidate solutions by the hundreds, these being then simulated for parameter prediction, and visualised as "clouds" of dots in the plane of power peaking and cycle length. Its successor, FUELGEN (Zhao, 1996 sqq), applies evolutionary computing, again by hyperrecursion. Arguably anticipation—and thus hyperincursion because of its being joint with hyperrecursion—apply to both FUELCON and FUELGEN: during the operation cycle (other than at downtime periods), state observability not obtaining for such variable which require direct inspection, current state estimates are based on those parameters which are observable, along with the forecast that was obtained by simulation at downtime, so that current state estimates partly depend on future states.

Keywords: nuclear power plants, refuellings, reload pattern design,
in-core management problem, heuristic problem-solving for allocation problems.

1 Introduction: The Context of the Project

The consecutive waves of methods from computing: linear programming from operations research; then, expert systems based on heuristic rulesets, and other knowledge-based systems from artificial intelligence; and then again, artificial neural networks; next, genetic algorithms, and finally, hybrid methods (Nissan, 1998e) combining the above, have had a major impact in nuclear engineering over the last two decades, and especially in the domain inside it with which we are concerned in particular, namely, computational methods for designing refuellings. This is also called: solving the in-core fuel management problem (as opposed to the management of fuel kept in store). As energy is produced, fuel becomes depleted; yet, the degree of depletion depends, among the other things, on whether the fuel units considered were entered as part of a fresh fuel batch, or were being reused, at the start of the relevant cycle. In fact, most nuclear plants operate by cycles,

International Journal of Computing Anticipatory Systems, Volume 9, 2001
Edited by D. M. Dubois, CHAOS, Liège, Belgium, ISSN 1373-5411 ISBN 2-9600262-2-5

whose length is variable, but which are about one year long. A standard textbook on the nuclear fuel cycle is Cochran and Tsoulfanidis (1990); see especially Sec. 6 on our present topic.

Bernard and Washio (1989) surveyed ongoing projects applying expert system technology to nuclear engineering. For refuelling in particular, two kinds of tasks have had expert systems or prototypes deal with them:

- the repositioning of fuel-units based on the physical constraints of a crane having to take out, transfer and reinsert the unit in another position in the grid which represents the core,
- and designing such a reload of the fuel while deliberately ignoring such logistics of the material operation, while focusing on the benefits in terms of effective power peaking in the successive states of the reactor core, and of legth of the period over which the nuclear fission can be usefully sustained (for example, one year and a half instead of just one year), as well as (of course) in terms of safety (the power peaking being sustained, yet not excessive).

A project in the former category was NUCLEXPERT, described by Jardon and Dubois (1986). Quite possibly, an incentive for such a focus on the first version of the task, instead of the second option, may have been grounded in an approach to (or culture of) nuclear power as known from the French national context: the state has a monopoly, and moreover the provision of fuel and the management of nuclear power plants is integrated vertically, so that the national strategy is to have reactors refuelled on a strictly annual basis, with no attempt to achieve longer operation periods between successive shutdowns of the reactor. Whereas vertical integration also obtains in the United States, managers of the individual plants there (as opposed to the company-level decision-making typical of the United Kingdom) have the interest of increasing the temporal distance between shutdowns, apart from the generalized requirement that downtime periods be kept short (they usually take a few weeks). This is not to say that in the United Staes there has been no attempt to devise expert systems that would approach refuelling from the perspective of planning crane movements. "Expert systems have also been developed to assist with scheduling the movements of fuel assemblies. One such system is CLEO, which was developed at Hanford for use in the Fast Flux Test Facility [(Smith, 1988; Smith et al., 1985)]. Also, EPRI, in conjunction with Intellicorp and the Virginia Electric Power Company, has developed a prototype expert system [(Naser et al., 1987; Colley et al., 1988)] that plans crane movements for the fuel insert shuffle of a PWR" (Bernard and Washio, 1989, p. 41). Samary Baranov, a formerly Soviet researcher, at one time "developed—for application at Soviet nuclear plants—a method by which a solution algorithm is converted into an optimized logic circuit design for incorporation into the hardware. (Baranov's conversion procedure is independent of the application domain, and was exemplified in several ways, in a recent book (Baranov, 1994), which however does not include the by now old, but as yet unpublished application to nuclear power plants)" (Nissan, 1998a). Arguably the earliest project in the second category—the one which tries to achieve good solutions in the plane whose two dimensions are power peaking and cycle length—were FUELCON (Galperin and Nissan, 1988), as well as a prototype whose ongoing development had been reported in Rothleder et al. (1988) and previously in Poetschhat et al. (1986), and Faught (1987). Tahara (1991) reported about an expert system for reshuffling, i.e., like the previous project but unlike FUELCON, emulating the way a human expert would try to generate an admissible and hopefully good fuel allocation into the positions of the grid representing the reactor core; FUELCON, instead, generates many candidate fuel-configurations at once and, which is what matters in the above respect, ex nihilo. Another tool for reload design is AUTOLOAD, described by Li and Levine (1994); in AUTOLOAD, an expert system with heuristic rules is integrated with search by means of the SCAM-W algorithm.

Other publications on FUELCON include Shuky Kimhi's thesis (1992), as well as the papers: Galperin et al. (1989, 1993), Galperin and Kimhi (1991), Galperin (1995), Nissan and Galperin (1998), Nissan (1998d), Galperin and Nissan (1998). whereas an attempt to incorporate automated, neural revision of the ruleset was announced in Galperin and Nissan (1989), and then reported in: Galperin et al. (1995, 1998), Siegelmann et al. (1997), Nissan et al. (1994a, 1994b, 1997).

FUELGEN, which adopts a genetic-algorithm based approach to refuellings, was the subject of Jun Zhao's thesis (1996); it has been described in the papers: Zhao et al. (1997, 1998a, 1998b), Nissan et al. (1998), Soper (1998). Other attempts to apply genetic algorithms to refuelling include the ones reported in: DeChaine and Feltus (1995, 1996), Poon (1992), Parks (1995, 1996). Nissan (1998a) is an overview of expert system and neural network applications to nuclear power systems. Nissan (1998b) is a journal special issue on the same subject (with also a paper on electrical power systems, and another paper on a neuro-fuzzy tool for nuclear power plants). Nissan (1998c) is a journal thematic section comprising five papers on FUELCON and FUELGEN — an accolade which bears witness to the visibility of this suite of projects, on which see also Nissan et al. (2000).

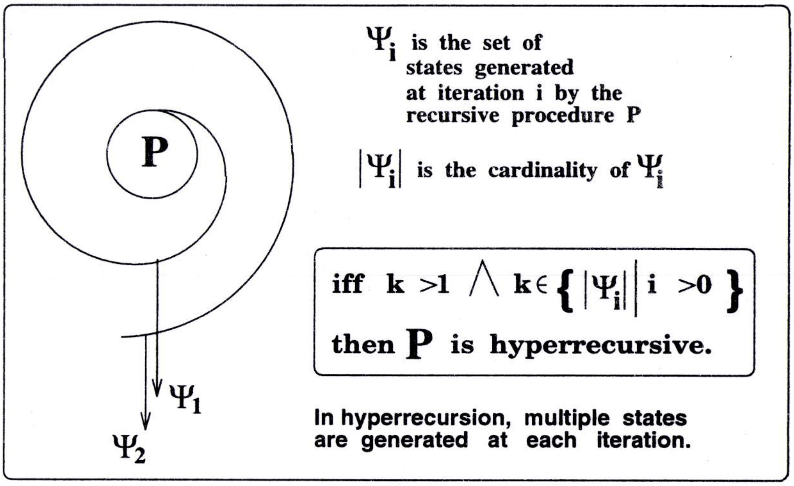


Figure 1: A schema of hyperrecursion.

2 Recursion, Multiple States, and Anticipation

Hyper-recursion takes place when, generally speaking, multiple states are generated at an iteration of a recursive procedure, instead of merely one state obtaining. See **Figure 1.** As we are going to see, both FUELCON and FUELGEN are hyper-recursive indeed. Their unifying feature is that they generate several alternative candidate fuel-configurations, which are then simulated, visualized, and thus ostensibly contrasted as to their respective benefits. Multiple state generation at each recursive state is, in that sense, one of the things which set FUELCON apart, as soon as it appeared, from extant tools for computer-assisted refuelling.

Anticipation permeates several tasks and functions in many a domain. In particular, in a computational anticipatory system, the current state estimate depends (also, or even: only) on future states. When anticipation co-occurs with hyper-recursion, we have hyper-incursion, as shown in **Figure 2.** Refuellings being designed with future performance of the nuclear reactor in view, has much to do with forecasting, for all of this not being robust, as we are going to see. Parameter prediction by means of simulation comes downstream of the generation of candidate fuel-configurations—this being the case in both FUELCON and FUELGEN. Moreover, state observability being much diminished (notwithstanding the monitoring of some parameters) while the fuel is burning, direct inspection is possible only at downtime periods. Unless the reactor is shut down, there are aspects of current state estimation which depends on the previous simulations of the operation cycle, and therefore (in part) on states which are still future. Arguably, FUELCON and FUELGEN are hyper-incursive. We are going to take a closer view of why this is the case indeed.

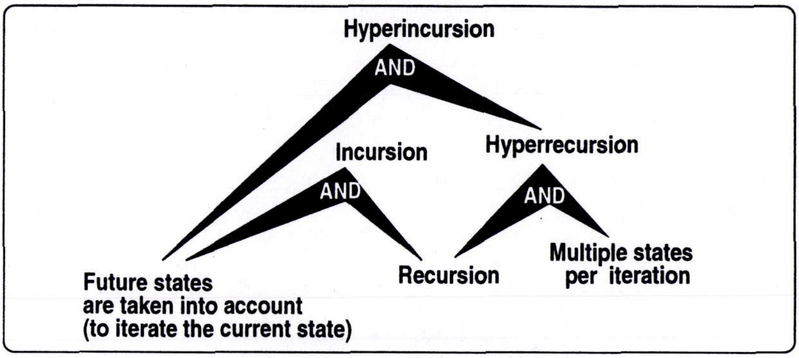


Figure 2: Hyperincursion versus hyperrecursion.

3 Fuel Reload Pattern Design, for Refuelling a Nuclear Reactor

Nuclear power plants include reactors. We are going to be mainly concerned with pressurized water reactors (PWR) in this paper; the FUELCON and FUELGEN projects cater to this kind of reactors. Viewed from the outside, the vessel of a reactor is not much unlike that of a water boiler. In the same building as the reactor, other, more slender boiler-like containers are found, towering around the vessel, and housing are the steam generators, acting as heat exchangers. (Yet, in the European Pressurized Water Reactor, or EPR, heat exchangers are horizontal.)

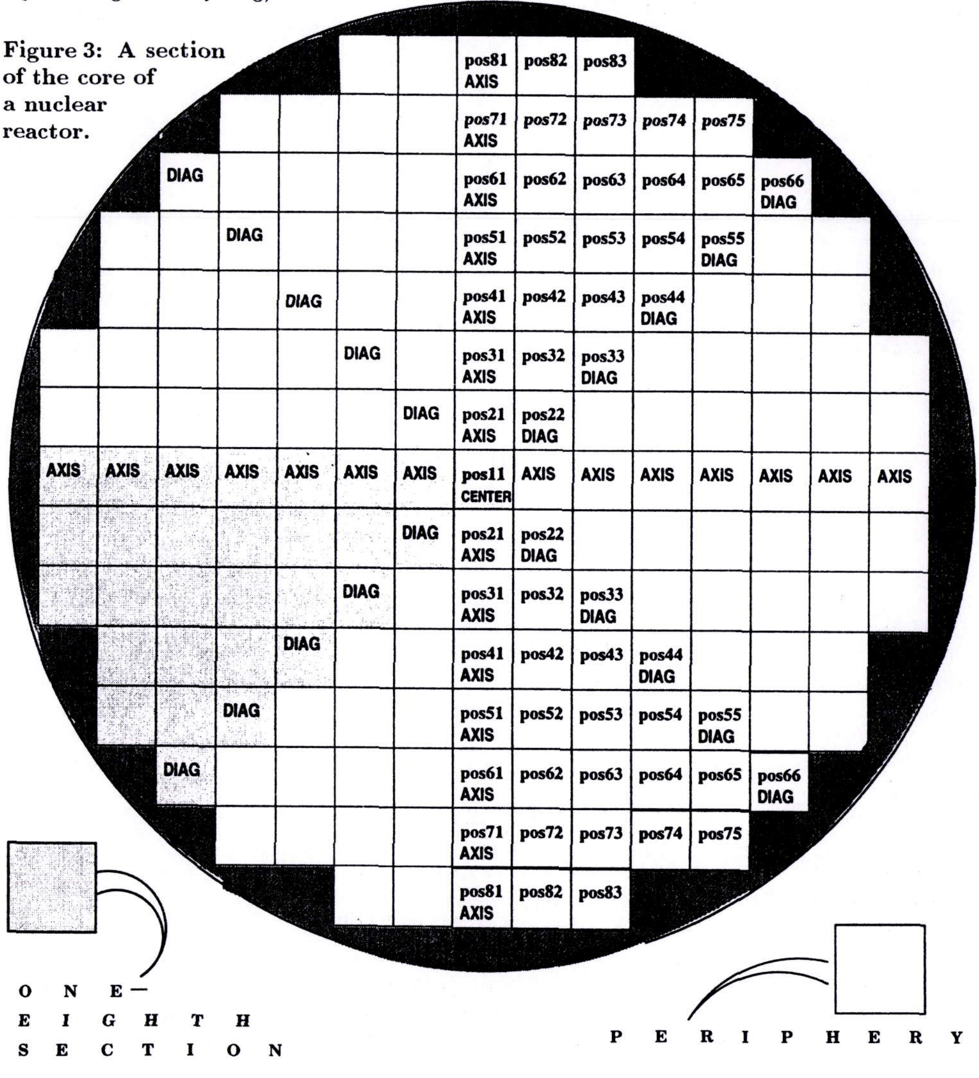
Inside the reactor vessel, the core barrel contains the reactor core proper. Nuclear fuel comes in the form of assemblies of 200 to 250 parallel rods, whose typical length is 350 cm. Fuel assemblies (for our purposes, fuel units) are inserted vertically into the core, which for the purposes of designing the refuelling can be viewed as a planar horizontal section, as shown in **Figure 3.** It is practically sufficient to reason on a slice of the grid, in one-eighth (or, less often, one-quarter) symmetry; hence, *pos11* labels the core position at the center, but the rest of the labelling in the same figure is in one-eight symmetry, for the purposes of just one slice. The upper tip of the slice is the center of the core. In the core geometry, important regions include the border of the core, the main axis, the diagonal, as well as the core periphery. All of these are indicated in **Figure 3.**

One fuel-unit is inserted by being lowered vertically into a given square position in the grid; all core positions are to contain one fuel unit. By "square", such a position is intended that it has four and only four contiguous positions in the same grid, unless the position is on the periphery of the core. (Actually, a different kind of core geometry obtains in former-Soviet reactors, where contiguity is between hexagonal positions in the grid. Moreover, in the Canadian deuterium uranium (CANDU) reactors, fuel is inserted horizontally; but for that matter, the kind of fuel reload design under discussion in this paper is not relevant for CANDU reactors, because for one thing, there is no need to shut down the reactor in order to load there the fuel units: natural uranium is used as fuel, and there is no need for downtime periods (at least, for the purposes of refuelling), as replacement is on-line, i.e., continuous.

Let us go back to a conventional, PWR context. A few of the positions in the core are to house not only a fuel unit, but a control rod as well, which in turn is to be lowered into the position in order to stop the nuclear fission reaction and shut down the reactor. A reactor is to be stopped either for its regular shut-down period (as intended for refueling and maintenance), or because of an emergency. To avoid overheating, liquid coolant (such as pressurized water in the PWR kind of reactor) is necessary, among the fuel rods. In pressurized water reactors, pressure is intended to prevent the water from boiling; there also exist boiling water reactors, or BWR. "In a conventional pressurized-water reactor (PWR), like the ones at Three Mile Island, water flows through the core, where it is heated under pressure of 2200 pounds per square inch. The water flows through a steam generator, where it turns a second closed loop of water to steam. The steam powers turbines that drive the electric generators. A boiling-water reactor (BWR) operates in a similar way, except that water in the primary loop is only at 1000 pounds per square inch, and so itself turns to steam that is piped directly to the turbines" (Fischetti, 1987, p. 31).

The nuclear fuel cycle includes not only the positioning and depletion of the fuel inside the reactor, but, as well, those stages that take place upstream and downstream of the fuel burning inside the reactor core, from the extraction of the natural uranium ore, through enrichment and fuel assembly manufacturing, down to the disposal of the spent fuel once it is discharged from the reactor (unless it is recycled, where this is permitted: in 1977, the U.S. government banned fuel reprocessing and recycling).

Figure 3: A section of the core of a nuclear reactor.



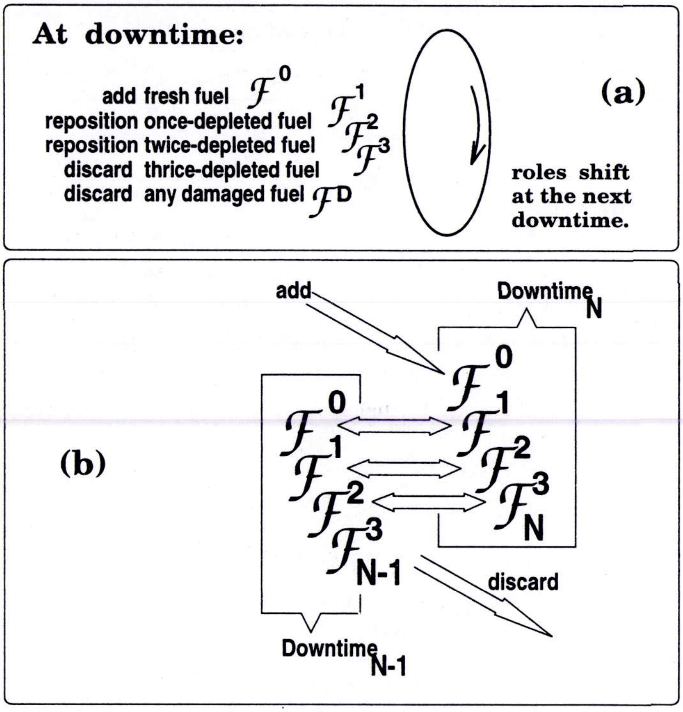


Figure 4:
A notation for batches of fuel.

Importantly, this back end of the process is not coincident with periodical shutdown and refueling: along with the three batches of fuel units (fresh, one-burnt, and twice-burnt) that are available for use in the design of the refuelling, there is just one batch (thrice-burnt fuel) that is to be discarded, apart from such fuel units that upon being inspected, are found to be damaged. The fuel units from the two other batches are simply to be repositioned, according to the suitability of their respective degree of depletion. For example, fresh fuel is not to be positioned at the centre of the grid, to avoid too high a localized power density, which could damage the fuel units (or, worse, result in hazard in respect of safety). **Figure 4(a)** introduces a notation for the various batches of fuel, along with the operation which is respectively relevant. In order of recency of inclusion into the pool, it's the two batches at the extremes that are concerned by, respectively, addition into the pool, and removal from further use: see **Figure 4(b).** At each given regular shut-down out of a sequence, the batches are more depleted, and move down in the sequence as shown in **Figure 4(b).** Some fresh fuel is added, and some of the older fuel is discarded: see **Figure 5.**

4 Forecasting, its Limitations, and Target Power Distribution

A typical PWR core generates, by 157 assemblies, about 3000 MW (megawatt)—these being thermal—from which about 1000 MW (electric) are produced. A single fuel rod generates about 80 KW (kilowatt) of thermal energy. One fuel assembly, consisting of 200 to 250 rods, generates about 20 MW (megawatt). Dontime periods, when the reactor does not produce power, are costly, and plant managers would prefer to keep them short. One would think it to be desirable for the design of the next refuelling to be ready for the refuelling itself to be carried out immediately as the reactor is shut down; yet, this is not possible. There is an unavoidable bottleneck. Inspection is necessary, and then the design of a suitable fuel reload may take a few days to accomplish, because not just the design proper, but simulations as well must be carried out. In fact, once the

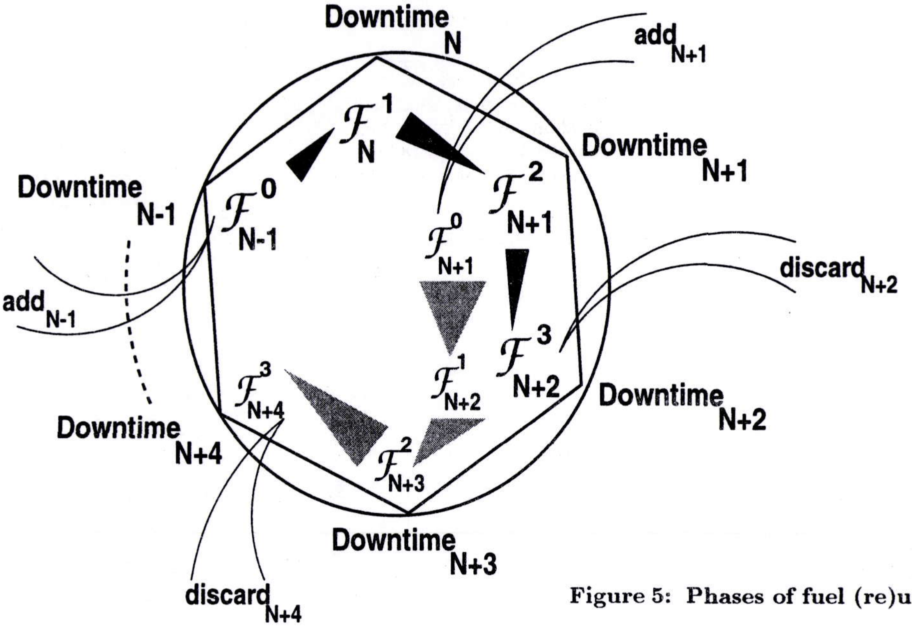


Figure 5: Phases of fuel (re)use.

new downtime period arrives, i.e., at end-of-cycle (EOC), the fuel manager could not possibly have a ready-made design for reload, i.e., a tailored, reasonably good configuration that assigns one fuel-assembly (of a given kind) to each single position in the core geometry.

"To account for that, it is not enough to point out that each power production cycle is unique; it is, indeed, because of specific power demands, availability of fresh fuel, past cycle length as yielding certain cumulative burnup values for each fuel-assembly, and expected length of the next cycle. The impossibility to prepare the reload design beforehand, stems from the fact the actual situation at each EOC is *not predictable* to any useful degree of exactitude. During the cycle, the fuel becomes depleted, and tentative forecasts of future states are possible —and simulations are indeed performed—but the final state of the fuel (and of the resulting power densities) is known accurately enough only at EOC, or a few weeks ahead. Uncertainties are such that they invalidate predictions of over two or three months ahead. Were a design prepared for the next cycle of the particular reactor, as based on forecasts for the end of the current cycle, that solution would not be *robust* enough in front of unforeseen variations, to fit the actual situation at EOC" (Nissan and Galperin, 1998). "It is necessary to reach EOC (or, often, to be about to reach it), as an enablement for assessing EOC state and, thus, producing the design for the new cycle. This depends on sensitivity to variations such as irregularities in power supply (stops, or increases that had to be enacted hastily to satisfy sudden demand)" *(ibid.)*.

Factors and their dependency as involved in the "noise" affecting predictions are described in **Figure 6.** Such factors "may cause forecasts for the state of the reactor core at the next EOC, not to correspond to its actual state at EOC. Uncertainties include (but are not limited to) the following: *(a)* network requirements, depending on the availability of other functioning reactors ("units") in the same plant; *(b)* the availability of other plants (in face of possible technical problems); *(c)* climatic conditions that influence power consumption; and *(d)* the availability of reload-maintenance teams at EOC. The latter is a factor that may shift the reactor shut-down by some weeks" *(ibid.)*. "Downtime periods at plants typically take a few weeks per years, and are costly, in

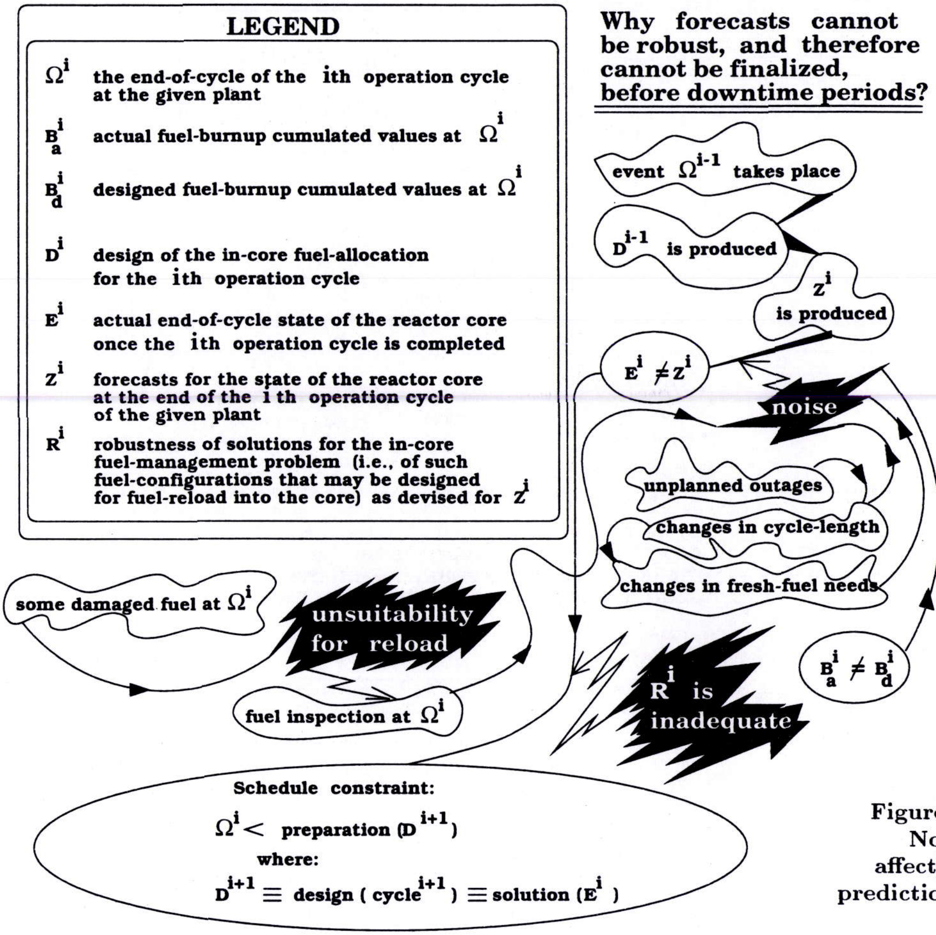


Figure 6: Noise affecting predictions.

terms of energy not supplied (and the need, for a company, to provide an alternative supply). Good fuel-allocation design strives for efficient peaking during the fuel cycle, and for as long a cycle as possible, but shortening downtime periods is also a goal. Cutting down inactivity periods by just planning ahead before reaching EOC is tentatively possible, and experienced engineers actually do, but notwithstanding a flexibility window, such forecasting is not robust enough for a ready-made design to be applied to the problem at hand the way it results to be at EOC. Indeed, the length of the cycle may result to have been longer or shorter than expected, e.g., because of climatic conditions intervening in place or elsewhere having determined levels of supply different for those planned, because another reactor or plant having been shut down required the plant at hand to provide alternative supply, because of faults, etc. Also, at EOC it may be the pool of available fuel will be different than expected, because some fuel-assembly was found to be damaged and needs replacement" (Nissan, 1998a). "Whereas in the published literature of in-core fuel management, real case studies are discussed, the situation for which the given solutions are suitable is not the same as the unicum constituted by your own plant at the end of a particular cycle" *(ibid.).* In such countries where state monopoly makes it affordable to renounce the achievement of as long cycles

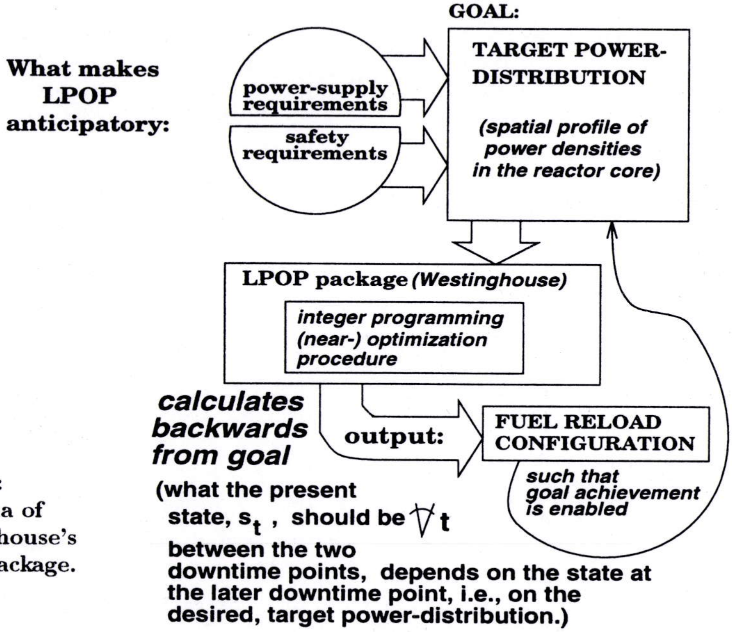


Figure 7:
A schema of Westinghouse's LPOP package.

as possible and to seek uniformity, instead—namely, such being the case of France—it has been possible to maintain a database of thousands of past solutions, with the expectation that enhanced interchangeability among reactors may make it possibly to match an allocation problem currently at hand against the experience of the past. Such pattern matching, or even case-based reasoning, is not as feasible in the industrial and institutional realities which are relevant elsewhere. Tools such as FUELCON and FUELGEN have been devised in order to cope with such heterogeneous contexts, where the limitations of forecasting and of available pools of real-case studies make it preferable to embody the expert's heuristic rules (as in FUELCON) or suitable finetuned parameters for evolutionary computing (as in FUELGEN) in a tool which generates a multitude of alternative solutions, to be contrasted by predicting the parameters of how they would perform if adopted.

Parks and Lewins (1992) outlined three categories of computerized tools for in-core fuel management: *manual design packages* (the engineer's own expertise is applied, and software only assists with calculations for analyzing the reactor core physics, and for visualizing the data): *expert systems* (which, like FUELCON, have a ruleset prune the search space, seeking good configurations): and *optimization packages*. One such packages is Westinghouse's LPOP, described by Alsop et al. (1991), and whose conceptual structure we illustrate in **Figure 7.** LPOP adopts an integer programming (near-)optimization procedure that starts from a target power distribution and calculates backwards how to obtain it, by matching the available fuel inventory. The next section describes the structure of FUELCON and FUELGEN.

5 FUELCON versus FUELGEN

A set of predefined heuristic rules is executed recursively by FUELCON's control component: see **Figure 8.** The ruleset comprises a few rules, "elimination rules", which constitute mandatory avoidance constraints (to prevent too high local power densities in the reactor core), and a few more rules, "preference rules", which vary according to the performance-seeking strategy to be adopted. Moreover, the loading sequence of the fuel assemblies is also heuristically predefined, and is a given, in the same way the rules are. Such heuristic knowledge is applied to the data in the

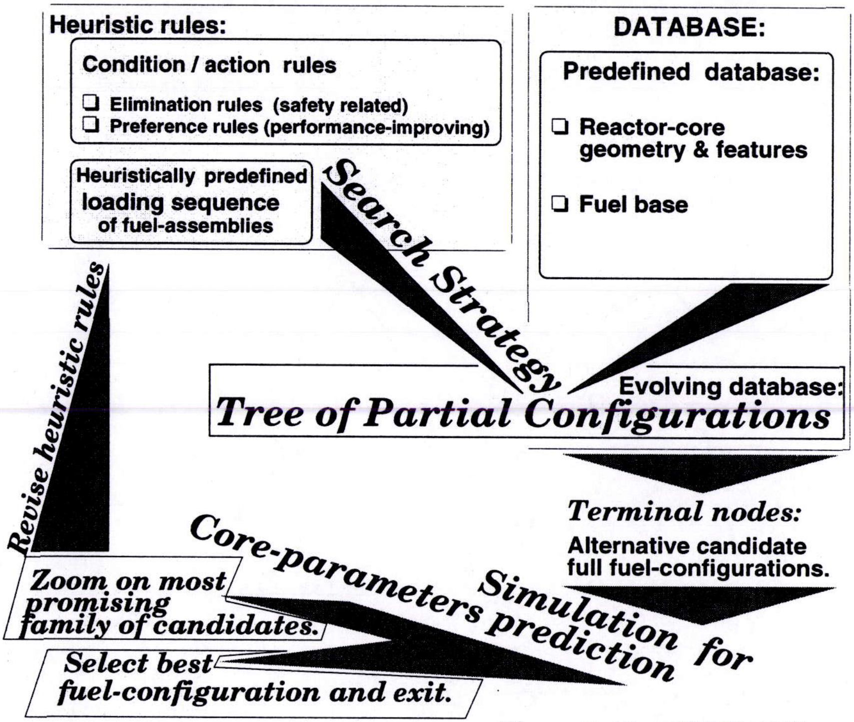


Figure 8: How FUELCON executes.

stable database of the problem at hand (the geometric structure of the core's symmetric slice, the positions at which the control rods are located, the pool of available fuel assemblies). The ruleset is executed, determining which nodes to generate (or, instead, avoid generating) as, level by level, a tree is being built: see **Figure 9.** The tree levels correspond to a given fuel-unit out of the given loading sequence. The terminal nodes will be alternative fuel configurations (i.e., fuel reload patterns). Nonterminals are partial configurations. At any given level in the tree, the sibling nodes display the grid of the reactor core; those fuel-units corresponding to tree-levels which are upper than the current level, appear to be positioned as inherited, i.e., in the way selected for the relevant ancestor-nonterminal of the given node (i.e., each node inherits the partial configuration having been constructed along the path leading to the node itself). Moreover, the fuel-unit corresponding to the given tree-level is positioned differently in those nodes (of the same given level) whose immediate ancestor is the same. The solution-space generator is based on the beam-search algorithm, thus, we have there a combination of the best-first and breadth-first methods. Solution space expansion is performed in the tree, and, at each level, several best branches are developed; the rest of the branches are discarded, which makes backtracking impossible.

Typically, the user is an experienced fuel manager, who will further want to improve upon the solutions yielded by the first round of ruleset execution, once they are simulated. The user can do that by modifying the ruleset: to steer the strategy he or she would refine the "preference rules", for example to achieve a zooming effect on a region in the space of solutions) This forms a loop of automated computation and manual intervention, and each round of executing the ruleset, recursively generates hundreds of terminal nodes in the tree constructed in the working memory

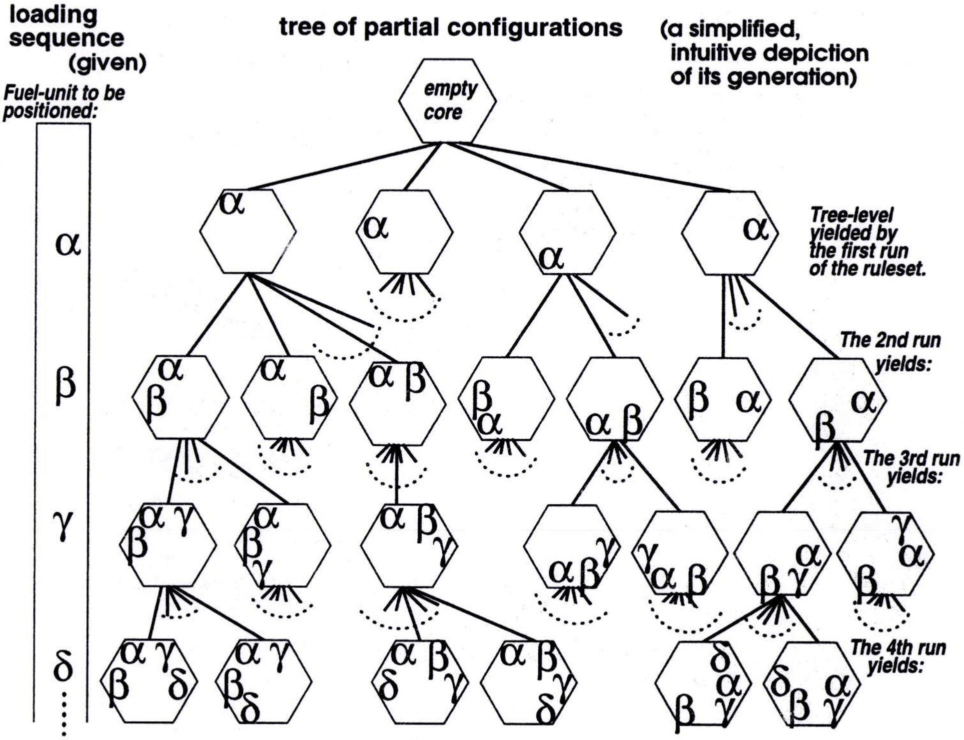


Figure 9: FUELCON's tree of partial configurations.

of the expert system. These hundreds of candidate solutions are projected in the plane of power peaking and cycle length, and are visualized as dots in the admissible or forbidden regions of the plane; see **Figure 10.** The simulation predicts the dynamics of the reactor physics while the fuel (arranged as per the given configuration) will be burning up to the next EOC.

In contrast to the transparency of FUELCON's ruleset, its functional replacement, FUELGEN, works like a black box, and is more suitable for such fuel managers who are not necessary as experienced as FUELCON's users. FUELCON is like a sports car, which does not suits everybody who wants to drive a car. FUELGEN adopts MacroGA, the "macro genetic algorithm" we developed, within the framework of what is known as macroevolution, in genetic algorithms research. In macroevolution, a number of species co-evolve. In FUELGEN, we choose not to have dominant species among these. Migration is controlled by a migration rate (or probability) determining when migration occurs, the number of migrants per migration (normally two), and a migration topology, such as the so-called ring or line topology. Migrants are the best individuals in the source species, and they will compete with all the individuals within the destination species for survival; the same number of least fit individuals will then be removed. Each species has a different crossover and mutation rate, which are generated from a mutation rate range and a crossover rate range. The ranking selection method is applied to all species. It is only as a particular case, that all species may happen to use the same crossover and mutation operators. Once in the macro genetic algorithm the parameters and a set of species are initialized, the probability of migration is set, and each species is initialized in terms of selection operator, crossover operator, mutation operator, population size,

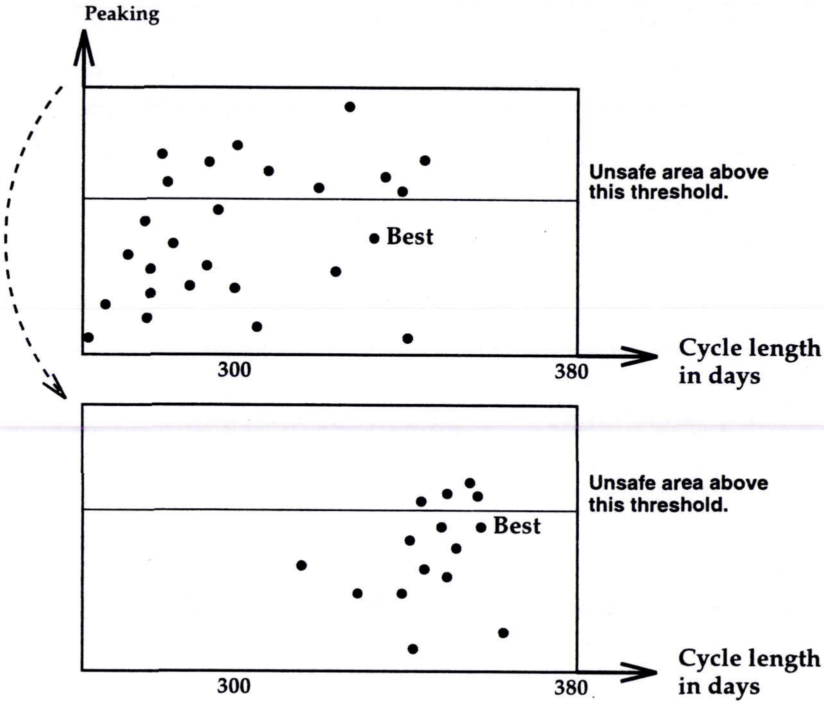


Figure 10: Improved output sets of candidate refuellings.

and crossover and mutation rates. Afterwards, all species evolve in parallel, and migrations take place at each generation. As each generation is created, the fitness for each individual in the population is evaluated, and the best individual is recorded and checked; next, the genetic operators are applied, in order to create a next generation. Yet, if the current best individual is acceptable in terms of fitness, (i.e., if a fuel-loading pattern has been found that is as good as we want it to be), then a termination message, "Stop", is distributed to all species, and the macroevolution process comes to an end. See **Figure 11.** One of the advantages is that we have a set of weakly interacting species, allowing each species to concentrate on a separate area of the search space. The species can search their own regions aggressively (under high selection pressure) and hence quickly, without the risk of a premature loss of diversity. Moreover, as we allow the use of different crossover and mutation rates in the various species, we avoid the problem of having to determine effective values for these parameters.

FUELCON, and FUELGEN as well (because of its checks on populations), exhibit the advantage that when parameter-predicting simulation is carried out, it reckons with, and plots, hundreds of solutions instead of just one. Moreover, a major feature of both FUELCON and FUELGEN (a feature which sets them apart from other projects reported in the specialized literature) consists of the fact that the fuel reload pattern design is integrated with the inclusion of the so-called "burnable poisons" (BP), instead of only incorporating these (and recalculating) downstream, at a later phase of the problem-solving process. This is a definite advantage, in that the fuel-allocation problem is solved similarly (and at the same go) when the design specifications for solving the in-core fuel management problem for pressurized water reactors is with or without BPs. The latter are rods of neutron absorbers inserted in part of the fuel assemblies, to better control reactivity.

What makes the two tools hyperincursive? Let us notate as $\mathcal{O}_\rho$ the set of observable states

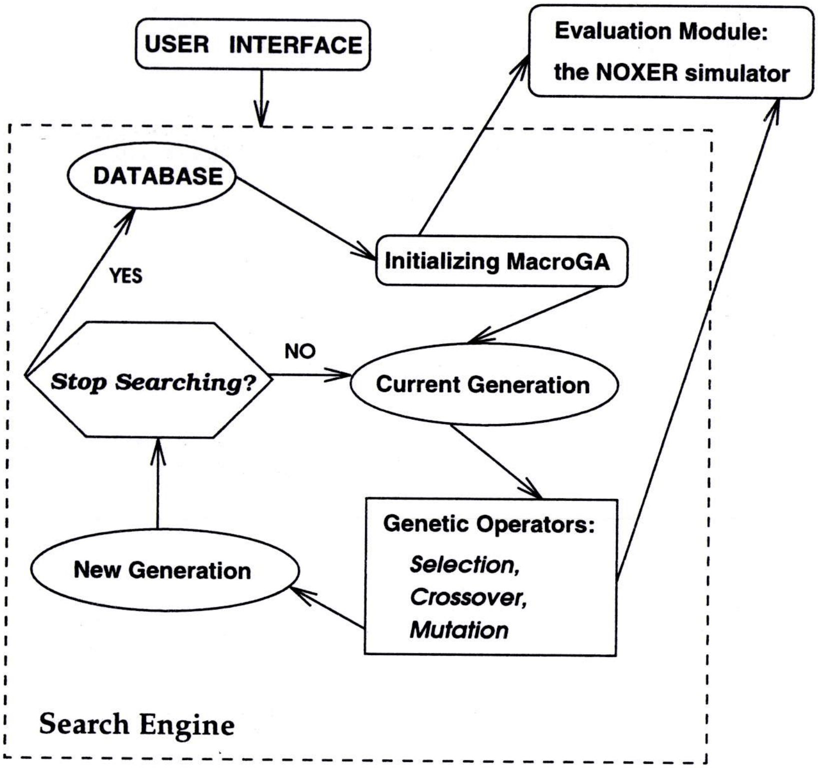


Figure 11: The structure of control in FUELGEN.

of reactor ρ, and as Π_ρ the set of parameters of the same reactor ρ. Let such parameters be indicated as $x(t) \in \Pi_\rho$. (Only parameters that can be at least sometimes observed are included in Π_ρ, by our definition. And then, let S_A be the set of (practically) possible observations (i.e., of observable states) of the parameters at the jth occurrence of a downtime period:

$$S_A \equiv \left\{ x(t_{\text{Downtime}_j}) \mid x(t) \in \Pi_\rho \right\} \qquad \text{where:} \qquad S_A \subset \mathcal{O}_\rho .$$

Next, let as notate as S_B the set of observable states while the fuel is burning. that is to say, between two successive downtime periods:

$$S_B \equiv \left\{ x(t) \mid t_{\text{Downtime}_j} < t < t_{\text{Downtime}_{j+1}} ,\ x(t) \in \Sigma_\rho(t) \ ,\ \Sigma_\rho(t) \subset \Pi_\rho \right\}$$

where: $S_A \bigcup S_B = \mathcal{O}_\rho$.

The subset of reactor parameters that are observable even between two successive downtime periods is considerably smaller that the set of all parameters that can be observed at least sometimes:

$$|\Sigma_\rho(t)| \ll |\Pi_\rho(t)|$$

Now, fuel reload pattern design is carried out and simulated at t_{Downtime_N}, for $\boldsymbol{t}$ such that $t_{\text{Downtime}_N} < t < t_{\text{Downtime}_{N+1}}$.

$\boldsymbol{S_C}$ is that subset of the states of that simulation that are future with respect to any time $\boldsymbol{t_K}$, such that $\boldsymbol{t_N} < \boldsymbol{t_K} < t_{\text{Downtime}_{N+1}}$, when the team at the plant is estimating the current state, $\boldsymbol{x(t_K)}$, of the reactor $\boldsymbol{\rho}$, based on $\boldsymbol{S_A}$, $\boldsymbol{S_B}$, and $\boldsymbol{S_C}$. This makes for anticipation, and inasmuch as the design is carried out by using either FUELCON or FUELGEN, the process of making use of the tool is anticipatory. As we have already seen, FUELCON and FUELGEN are hyperrecursive, so hyperincursion obtains.

6 Conclusion

FUELCON and FUELGEN, two expert systems for the design of fuel reload configurations in nuclear power systems, constitute the outcome of two different phases in a suite of projects. FUELCON is rather obviously hyperrecursive, because of how, during a session, the tree of partial alternative configurations is generated. FUELGEN is driven by a genetic algorithm, and the generation of successive generations is somewhat trivially amenable to hyperrecursion. Moreover, anticipation makes these tools hyperincursive.

Acknowledgments

We are grateful to Shuky Kimhi, Meir Segev, and Hava Siegelmann for their role at various stages of this long project.

References

B. H. Alsop, Y. A. Chao, B. J. Johansen and T. Morita (1991) Evaluation and Performance of Westinghouse Loading Pattern Search Methodology. *Proceedings of the International Topical Meeting on Advances in Mathematics, Computations, and Reactor Physics,* Pittsburgh, PA: Part 5, 22.2, 2–1 ff.

J. A. Bernard and T. Washio (1989) *Expert Systems Applications Within the Nuclear Industry.* American Nuclear Society, La Grange Park, Illinois.

R.G. Cochran and N. Tsoulfanidis (1990) *The Nuclear Fuel Cycle: Analysis and Management.* American Nuclear Society, La Grange Park, Illinois.

R. Colley, J. Naser, J. Gaiser and T. Brookmire (1988) Fuel Insert Shuffler: A Case Study in Expert System Development. In: D. Majumdar and J. Sackett (eds.), *Proceedings of the ANS International Topical Meeting on Artificial Intelligence and Other Innovative Computer Applications in the Nuclear Industry,* Plenum Press, New York, p. 625.

M.D. DeChaine and M.A. Feltus (1995) Comparison of Genetic Algorithm Methods for Fuel Management Optimization. *Proc. Int. Conf. on Mathematics and Computations, Reactor Physics, and Environmental Analysis,* Vol. 1, Portland, Oregon.

M.D. DeChaine and M.A. Feltus (1996) Fuel Management Optimization Using Genetic Algorithms and Expert Knowledge. *Nuclear Science and Engineering,* **124**(1).

W.S. Faught (1987) Prototype Fuel Shuffling System Using a Knowledge-Based Toolkit. Technical Report, IntelliCorp, Mountain View, California.

M. A. Fischetti (1987) Inherently Safe Reactors: They'd Work if we'd Let Them. *IEEE Spectrum* **24**(4): pp. 28–33.

A. Galperin (1995) Exploration of the Search Space of the In-Core Fuel Management Problem by Knowledge-Based Techniques. *Nuclear Science and Engineering,* **119**(2), p. 144 ff.

A. Galperin and S. Kimhi (1991) Application of Knowledge-Based Methods to In-Core Fuel Management. *Nuclear Science and Engineering,* **109**, pp. 103–110.

A. Galperin and E. Nissan (1988) Application of a Heuristic Search Method for Generation of Fuel Reload Configurations. *Nuclear Science and Engineering,* **99**(4), pp. 343–352.

A. Galperin and E. Nissan (1989) Discovery as Assisted by an Expert Tool: A Refinement Loop for Heuristic Rules in an Engineering Domain. *Proc. of the 16th Convention of Electrical and Electronics Engineers,* Tel-Aviv, 7–9 March 1989. IEEE, New York: Paper 1.4.3.

A. Galperin and E. Nissan (1998) The FUELCON Meta-Architecture, II: Alternatives for Parameter Prediction. In Ref. 25.

A. Galperin, S. Kimhi and M. Segev (1989) A Knowledge-Based System for Optimization of Fuel Reload Configurations. *Nuclear Science and Engineering,* **102,** p. 43 ff.

A. Galperin, S. Kimhi and E. Nissan (1993) FUELCON: An Expert System for Assisting the Practice and Research of In-Core Fuel Management and Optimal Design in Nuclear Engineering. *Computers and Artificial Intelligence,* **12**(4), pp. 369–415.

A. Galperin, S. Kimhi, E. Nissan, H. Siegelmann and J. Zhao (1995) Symbolic and Subsymbolic Integration in Prediction and Rule-Revision Tasks for Fuel Allocation in Nuclear Reactors. *Proc. 3rd European Congress on Intelligent Techniques and Soft Computing (EUFIT'95),* Aachen, Vol. 3, pp. 1546–1550.

A. Galperin S. Kimhi, E. Nissan and H. Siegelmann (1998) FUELCON's Heuristics, Their Rationale, and Their Representations. In Ref. 25.

P. Jardon and D. Dubois (1986) NUCLEXPERT: un progiciel d'Intelligence Artificielle pour l'optimisation des séquences de rechargement de centrales nucléaires. *Nouvelles de la Science et des Technologies,* **4**(2/3), pp. 99–102.

Y. Kimhi (1992) A Non-Algorithmic Approach to the In-Core Fuel Management Problem of a PWR Core. Ph.D. Dissertation, Nuclear Engineering, Ben-Gurion University of the Negev, Beer-Sheva. In Hebrew. [Kimhi's forename is "Yehoshua"; the form "Shuky" is also used.]

Z. Li and H. Levine (1994) AUTOLOAD, an automatic optimal Pressurized Water Reactor reload design system with an expert module. *Nuclear Science and Engineering,* **118,** pp. 67–78.

J. Naser, R. Colley, J. Gaiser, T. Brookmire and S. Engle (1987) A Fuel Insert Shuffle Planner Expert System. In: S. M. Divakaruni, D. Cain, E. Baytch and C. Saylor (eds.), *Proceedings of the EPRI Seminar on Expert Systems Applications in Power Plants.* Electric Power Research Institute, Palo Alto, CA, May 1987.

E. Nissan (1998a) Intelligent Technologies for Nuclear Power Systems: Heuristic and Neural Tools. *Expert Systems with Applications,* **14**(4), pp. 443–460.

E. Nissan (ed.) (1998b) *Intelligent Technologies for Electric and Nuclear Power Systems,* special issue, *Computers and Artificial Intelligence,* **17**(2/3).

E. Nissan (ed.) (1998c) *Forum on Refuelling Techniques for Nuclear Power Plants: One Decade with FUELCON.* Thematic section, *New Review of Applied Expert Systems,* **4,** pp. 139–194.

E. Nissan (1998d) The FUELCON Meta-Architecture, in the Landscape of Intelligent Technologies for Refuelling. In Ref. 25.

E. Nissan (1998e) Hybrid Techniques. [Five books reviewed.] *Computers and Artificial Intelligence,* **17**(2/3), pp. 251–256.

E. Nissan and A. Galperin (1998) Refueling in Nuclear Engineering: The FUELCON Project. *Computers in Industry,* **37**(1), pp. 43–54.

E. Nissan, H. Siegelmann and A. Galperin (1994a) An Integrated Symbolic and Neural Network Architecture for Machine Learning in the Domain of Nuclear Engineering. *Proc. of the 12th IAPR Int. Conf. on Pattern Recognition,* Vol. 2, pp. 494–496.

E. Nissan, H. Siegelmann, A. Galperin and S. Kimhi (1994b) Towards Full Automation of the Discovery of Heuristics in a Nuclear Engineering Project, by Combining Symbolic and Subsymbolic Computation. *Proc. 8th Int. Symp. on Methodologies for Intelligent Systems: ISMIS'94* (LNAI 869), Springer Verlag, Heidelberg, pp. 427–436.

E. Nissan, H. Siegelmann, A. Galperin and S. Kimhi (1997) Upgrading Automation for Nuclear Fuel In-Core Management: From the Symbolic Generation of Configurations, to the Neural Adaptation of Heuristics. *Engineering with Computers,* **13**(1), pp. 1–19.

E. Nissan, A. Soper, J. Zhao, B. Knight, M. Petridis (1998) Fuel Reload Pattern Design Within a Family of Hybrid Architectures. *Proc. FLINS'98: Third International FLINS Workshop on Fuzzy Logic and Intelligent Technologies for Nuclear Science and Industry (FLINS'98),* Antwerp, Belgium, pp. 408–415.

E. Nissan, A. Galperin, J. Zhao, B. Knight and A. Soper (2000) From FUELCON to FUELGEN: Tools for Fuel Reload Pattern Design. Ch. 20 in: D. Ruan (ed.), *Fuzzy Systems and Soft Computing in Nuclear Engineering* ("Studies in Fuzziness and Soft Computing" series, vol. 38). Physica-Verlag (Springer-Verlag), Heidelberg, pp. 432–448.

G.T. Parks (1995) Multiobjective PWR Reload Core Optimization Using a Genetic Algorithm. *Proc. Int. Conf. on Mathematics and Computations, Reactor Physics, and Environmental Analysis.* Vol. 1, Portland, Oregon.

G.T. Parks (1996) Multiobjective Pressurized Water Reactor Reload Core Design by Nondominated Genetic Algorithm Search. *Nuclear Science and Engineering,* **124**(1).

G.T. Parks and J.D. Lewins (1992) In-Core Fuel Management and Optimization: The State of the Art. *Nuclear Europe Worldscan,* **12**(3/4), p. 41.

G.R. Poetschhat, B.M. Rothleder, W.S. Faught and W.J. Eich (1986) Interactive Fuel Shuffle Assistant Graphics Interface and Automation for Nuclear Fuel Shuffle With PDQ7. *Proceedings of Topical Meeting on Advances in Fuel Management,* Pinehurst, NC, March 2–5, 1986, American Nuclear Society, La Grange, Illinois.

P.W. Poon (1992) Genetic Algorithms and Fuel Cycle Optimization. *The Nuclear Engineer,* **31**(6).

B.M. Rothleder, G.R. Poetschhat, W.S. Faught and W.J. Eich (1988) The Potential for Expert System Support in Solving the Pressurized Water Reactor Fuel Shuffling Problem. *Nuclear Science and Engineering,* **100:** p. 440 ff.

H. Siegelmann, E. Nissan and A. Galperin (1997) A Novel Neural/Symbolic Hybrid Approach to Heuristically Optimized Fuel-Allocation and Automated Fuel-Allocation in Nuclear Engineering. *Advances in Engineering Software,* **28**(9), pp. 581–592.

D. E. Smith (1988) Documentation of Knowledge in the Development of CLEO, a Refueling Assistant for FFTF. In: D. Majumdar and J. Sackett (eds.), *Proceedings of the ANS International Topical Meeting on Artificial Intelligence and Other Innovative Computer Applications in the Nuclear Industry.* Plenum Press, New York, p. 607 ff.

D. E. Smith, L. F. Kocher and S. E. Seeman (1985) CLEO: A Knowledge-Based Refueling Assistant at FFTF. *Transactions of the American Nuclear Society,* **50:** p. 292 ff.

A. Soper (1998) Exploring Genetic Alternative Concepts for FUELGEN. In Ref. 25.

Y. Tahara, K. Hamamoto and M. Takase (1991) Computer Aided System for Generating Fuel Shuffling Configuration Based on Knowledge Engineering. *Journal of Nuclear Science and Technology,* **28**(5): pp. 399–408.

E. Tanker and A.Z. Tanker (1995) Application of a Genetic Algorithm to Core Reload Pattern Optimization. *Proc. Int. Conf. on Mathematics and Computations, Reactor Physics, and Environmental Analysis,* Vol. 1, Portland, Oregon.

J. Zhao, (1996) An Examination of the Macro Genetic Algorithm and its Application to Loading Pattern Design in Nuclear Fuel Management. Ph.D. Dissertation, Computer Science, The University of Greenwich, London (discussed in November 1996).

J. Zhao, B. Knight, E.Nissan and A. Soper (1997) FUELGEN: A Genetic-Algorithm Based System for Fuel Loading Pattern Design in Nuclear Power Reactors. *Expert Systems with Applications,* **14**(4), pp. 461–470.

J. Zhao, B. Knight, E. Nissan, A. Soper (1998a) FUELGEN: Effective Evolutionary Design of Refuellings for Pressurized Water Reactors. In Ref. 24, *Computers and Artificial Intelligence,* **17**(2/3), pp. 105–125.

J. Zhao, B. Knight, E. Nissan, M. Petridis and A. Soper (1998b) The FUELGEN Alternative: An Evolutionary Approach. The Architecture. In Ref. 25.