Cooperative Methods for Security Planning

Sarosh Talukdar, V.C. Ramesh

EDRC 18-38-92

COOPERATIVE METHODS FOR SECURITY-PLANNING

Sarosh Talukdar V.C. Ramesh

Engineering Design Research Center
Hammerschlag Hall 2221
Carnegie Mellon University
Pittsburgh, PA 15213
Ph: (412) 268-8778
Fax:(412)268-5229

Fourth International Symposium on Expert Systems Application to Power Systems, La Trobe University, Melbourne, Australia, January 4-8,1993.

Cooperative Methods for Security-Planning

Sarosh Talukdar V.C. Ramesh
Engineering Design Research Center
Carnegie Mellon University
Pittsburgh, PA 15213
(412)268-8778
fax: (412) 268-5229

ABSTRACT

This paper develops a new approach to an old important and difficult problem: security planning in power system operations. The complexity of the problem is due primarily to two factors: (i) the large number of contingencies that need to be analyzed make it computationally intractable for real-time operations, and (ii) the problem has many solutions, and the global optimum is likely to be much better than the many local optima. We believe the problem is too big and difficult to be solved by any single, monolithic agent. Instead, we have developed a team of co-operative agents, called an A-Team, that is well-suited to solve this problem. The agents are autonomous, work in parallel, and communicate asynchronously. The paper describes the organizational structure of the team and presents results obtained both for the security-planning problem and for some other difficult global optimization problems.

Key-words: power systems security, parallel processing, distributed artificial intelligence, global optimization

1. PROBLEM FORMULATION

1.1 Background and Terminology

Think of a power system as a network containing m switches, each of which can be either open or closed. Thus, the system can adopt $M = 2^m$ different configurations, denoted by Co, C-|,... C M , where Co is the current configuration. Let X_n be a vector whose elements are the bus voltages and bus power injections of C_n. Though X_n contains both state and control variables, we will, for the purposes of brevity, call it a state vector.

The concerns in operating a power system can be divided into two broad categories: cost and quality. Cost is usually represented by a function: $f(X_n, D(t))$, where t is time and D is a vector of exogenous, time-varying quantities, such as customer demands for electric energy. Quality concerns are usually expressed as a set of nonlinear relations (sometimes, called load and operating constraints) that are configuration-specific, and have the general form:

$$G_n(X_n,D(t)) = 0$$
$$H_n(X_n,D(t)) \leq 0$$

X_n is said to be a normal state if it satisfies these constraints. S_n, the set of all normal states for configuration C_n, is called the normal set of C_n. Configurations for which S_n is empty are said to be uncorrectable; all other configurations are said to be correctable.

Two sorts of events can cause a system state to become abnormal: gradual changes in the exogenous variables, D, and sudden disturbances that result in random configuration changes. The latter can cause far larger excursions, and hence, are much more dangerous.

This paper is largely concerned with control actions that can be used to counter the effects of sudden disturbances. These actions can be discrete (switching operations) or continuous (changes in the independently controllable components of the state vector). We will concentrate on the latter.

Let $\tau_n(X_{na}, X_{nb})$ be the least time required to change the state of C_n from X_{na} to X_{nb} through a sequence of control actions. We will call τ_n a transition delay. Note that τ_n is non-zero because many control actions are rate limited. The output of a

typical generator can, for instance, be increased at most by a few megawatts per minute.

1.2 Optimum Power Flows (OPFs)

One of the simplest operating philosophies is to attempt to minimize instantaneous costs while keeping the state normal. In other words:

(OPF): Min $f(X_0)$

s.t. $G_0(X_0,D) = 0$

$H_0(X_0,D) \leq 0$

Since the dimensions of Xo, Go. and Ho are often of the order of 1000, this is a large problem; available techniques are barely able to solve it fast enough for the results to be useful in real-time operations.

1.3 Adding Contingency Constraints

How can one limit the ill effects of the random configurational changes that result from sudden disturbances? By far the most common practice involves two steps. First, a set of critical configurational changes (called contingencies) is identified. Second, plans are made to reestablish a normal state within some short period after each contingency.

The identification of critical contingencies requires system-specific knowledge, much of which can be encoded in expert systems [1]. In other words, much if not all of the identification process can be automated with existing techniques. The same is not true for planning responses to these contingencies. To understand why, suppose that the n-th contingency would cause the system's state to change from Xo to X_{nc}. If X_{nc} is abnormal, the planning problem is to find a normal state, X_n, that can be achieved within an acceptably short time, say T_n. There are two different ways to formulate this problem: the first treats correction times as hard constraints, the second, in a softer way, specifically, as terms of an objective function. The modifications that result to (OPF) from these two treatments are indicated below:

(CCP1): Min $f(X_0)$

s.t. $G_0(X_0, D) = 0$

$H_0(X_0, D) \leq 0$

$G_n(X_n, D) = 0$

$H_n(X_n, D) \leq 0 \quad n = 1,2 \ldots N$

$x_n(X_n - X_0) \leq T_n$

(CCP2): Min $f(X_0) + \sum_{i=1}^{N} w_n \tau_n(X_n - X_0)$

s.t. $G_n(X_n, D) = 0$

$H_n(X_n, D) \leq 0 \quad n = 0, 1 \ldots N$

where: N is the number of contingencies to be considered; T_n is the time allowed for the n-th contingency to be corrected, w_n is a weight assigned to the n-th contingency; and it has been assumed that $x_n(X_n - X_{nc})$ can be approximated by $\tau_n(X_n - X_0)$.

Both these formulations are very large-at least N+1 times as large as (OPF) As such, both are beyond existing capabilities for fast, reliable and repeated solution. In addition, each requires the user to select some parameters: $\{T_n\}$ in the case of (CCP-1) and $\{w_n\}$ in the case of (CCP-2). It happens that the selection of $\{T_n\}$ is much more difficult. The explanation is as follows. Let $X = [X_0, X_1, \ldots, X_N]$ be a vector called a super-state. Let S1 be the feasible set of (CCP1), that is, the set of all values of X that satisfy the constraints of (CCP1). Let S2 be the feasible set of (CCP2). Then S1 is small and sensitive to the values selected for $\{T_n\}$ while S2 is much bigger and insensitive to the values of $\{w_n\}$. Another way of putting it is that the constraints of (CCP1) require all the contingencies be correctable and also, ail the corrections be completed within time limits, $\{T_n\}$, that must be selected apriori. In contrast, the constraints of (CCP2) require only that all the contingencies be correctable. In selecting $\{T_n\}$ there is a considerable risk of making S1 empty, in which case little useful information may result from attempts to solve (CCP1), even though the attempts would be long and painful. In selecting $\{w_n\}$, however, the user is merely expressing an opinion on the relative importance of contingencies and could adjust this opinion interactively.

Because of (CCP1)'s profound disadvantages relative to (CCP2), we will henceforth consider only (CCP2). Also, recall that the vector of exogenous variables, D, is time varying, and therefore, the solution of (CCP2) is time varying.

1.4 A Decomposition

Notice that the constraints of (CCP2) consist of N+1 independent blocks. As a result, (CCP2) can be decomposed into a set. $\{(IP_n)\}$. of N+1 subproblems each having the form:

(IP_n): $\min_{X_n} f_n(X_n, \bar{X}_n)$

s.t: $G_n(X_n, D) = 0$

$H_n(X_n, D) \leq 0 \quad n = 1, 2, \ldots, N$

where:

$\bar{X}_n = X \backslash X_n$, that is, the super-state X with the elements of Xn removed

$$f_0(X_0, \bar{X}_0) = f(X_0) + \sum_{n=1}^{N} w_n t_n(X_n - X_0)$$

$$f_n(X_n, \bar{X}_n) = x_n(X_n - X_0) \quad \text{for } n = 1, 2 \ldots N$$

Let X* be a solution of {(IP_n)}. Then, it can easily be shown that X* is also a solution of (CCP2), and hence, {(IP_n)} and (CCP2) are equivalent.

1.5 A Skewed Approximation

Can the couplings among the members of {(IP_n)} be made more loose so their parallel solution becomes more easy?

Note that the exact solution of {(IP_n)} is unobtainable because the exact value of the exogenous vector, D, is unknown. The elements of D are time varying and are measured by sensors that can be hundreds of miles apart. There is always some delay and time skew in these measurements. What if delays and time skews were allowed for the values of X_n? More specifically, suppose that each (IP_n) is treated as a separate problem that is solved iteratively for X_n, while X_n and D are treated as exogenous variables whose values are updated as new estimates of them become available. Then we have a set, {(IPn')}. of more loosely coupled problems, each of the form:

$$(IP_n'): \min_{X_n} \; f_n(X_n, 5C_n)$$

$$\text{s.t:} \quad G_n(X_n, D') = 0$$

$$H_n(X_n, D') \leq 0$$

where X'_n and D' are the latest available values of X_n and D. Intuitively, one would expect the solution of (IP_n') to track the solution of (IP_n) as it varies in time, the tracking error increasing smoothly with the skew in the values of X'_n and D\ That this is actually the case is easily proved [2].

2. MULTI-AGENT SOLUTION PROCESSES

2.1 Asynchronous Teams (A-Teams)

The preceding sections have decomposed the contingency constrained problem, (CCP2), into a set, {(IPn^1)}, of N+1 smaller problems, each of the form and size of an optimum power flow. The smaller problems are very loosely coupled and can be solved by a team of agents working in parallel, provided the team is properly organized.

An organization can be characterized by a quadruple: (C, D, A, I); where: C is a graph, called a control flow, that shows who supervises whom; D is a graph, called a data flow, that indicates who does what and who may exchange data with whom; A is a set of criteria, called activity constraints, that prescribe how agents are to operate in time; and I is a set of criteria, called insertion constraints, that specify what must be done to add or delete an agent from the organization [3].

The space of all organizations contains a set whose members, called A-Teams, have two very desirable properties. First, they are exceedingly open (new agents can be added to an A-Team almost effortlessly). Second, they are easily distributed (an A-Team fits naturally into a network of computers, its agents use only locally available information and it is less sensitive to communication delays than other organizations).

An A-Team is defined as follows [3]:

- C, its control flow, is null, meaning that it contains no supervisors; all its agents are autonomous.
- D, its data flow is cyclic so its agents can use feedback and iteration in developing solutions.
- A, its set of activity constraints, is empty, meaning that its agents are free to act when they wish. In particular, there is no predetermined schedule for exchanges of information; rather, exchanges occur asynchronously (spontaneously). Moreover, all the agents can work in parallel all the time.
- I, its set of insertion constraints, is unspecified but tends to be "half empty." (Because the agents are autonomous, there is no managerial superstructure to modify when an agent is added or deleted; the only changes that need to be made are to the agent itself.)

Clearly, the structure of an A-Team allows for anarchic behavior. Autonomous agents, each deciding for itself what it is going to do and when, if ever, it will communicate its results, can act at cross purposes. Surprisingly, there are simple strategies, not only to prevent this from happening, but to make A-Teams high in performance (fast at finding good solutions to difficult problems). Two categories of these strategies are [3]: mixing and socialization. "Mixing" means choosing agents so there is a balance between those that create solutions and those that destroy them. The balance must be such that a population of solutions is maintained and herded along paths that lead to profitable conclusions. "Socialization" means the insertion of a few instincts (rules) in each agent that cause it to seek a local consensus (align its actions with those of its immediate neighbors).

Often, a well selected mix of agents is sufficient to make an A-Team effective, no special socialization strategy being needed. This seems to be the case in the contingency constrained problem, {(IPn^1)}.

2.2 A Data Flow

Consider (IP_n')- It uses one memory and five types of agents:

- Importers: collect values of X^*_n and D.
- Voyagers: coarse search

- Probes: fine search (conventional nonlinear programming algorithms)
- Inhibitors: place a "fence" around solutions that have been found. Fence's strength may decrease with time.
- Destroyers: eliminate obsolete solutions and fences

The voyagers need further explanation.

2.3 Coarse **Search** by a Relaxed Interior Point **Method**

(IP_n^f) is replaced by a vector field such that singularities of the field correspond to solutions of (IP_n'). The magnitude of the vector field is unity everywhere except at the solutions of IP_n' (where it is zero). The direction of the vector field is such that it points towards feasible regions when outside them, and towards minima when inside the feasible regions.

Voyagers circulate in this field influenced by the force exerted by the field. The force directly influences the direction of the voyager. The speed of the voyager is altered independently through heuristics; the speed is never zero. Consequently, the voyagers are always in motion (and don't stop at the singularities). The voyagers follow the flux lines in an inertial manner; thus are able to smoothly traverse areas where the field fluctuates rapidly.

The constraint-handling approach has the flavor of a relaxed interior-point method since the inequality constraints behave like elastic membranes permitting some violations but acting to reduce the violation by coercing the voyager back into the feasible region. In contrast, in a strict interior-point method, the constraints act as rigid walls permitting no violations. A relaxed approach has computational advantages since it is unnecessary to monitor constraints every iteration. Each equality constraint of the form G(X) = 0 is approximated by a pair of inequalities of the form: G(X) ≥ -a, G(X) ≤ a where a is a parameter.

The field leads the voyagers to the basins of attraction of the solutions of (IP_n'). In each basin, the Voyager plants a flag (Fig. 1) that serves as a launch-point for the Probes which conduct a fine-grained search of the basin to find the minimum. Heuristics govern the choice of this launch-point and are designed to minimize the chance of occurrence of more than one launch-point per basin. Also, when several basins (and hence, minima) are clustered close together, the heuristics try to select a single launch-point in the basin of the best solution, thereby avoiding the determination of inferior solutions.

Repeated re-determination of found solutions is avoided through the erection of "fences" (spherical constraints) around known minima. These fences are one way in which the voyagers help each other out by preventing others from revisiting regions visited by one.

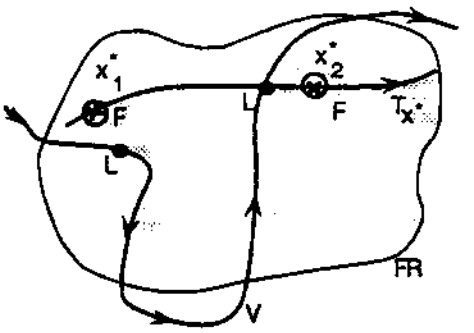

Fig. 1: V represents the trajectory of the voyager; T_x* is the trajectory of the solution x\ x*i, x*2 are the solutions obtained at times tt, \2- FR represents the feasible region, L denotes launch-point, and F is a fence.

3. **EXAMPLES**

This section presents the results of using the A-Team approach both for the security-planning problem in power systems, and for some other difficult global optimization problems.

3.1 Power Systems

A system with 6-buses, 11-lines, and 3-generators [4] was used to test our approach to the CCP2 problem. A single line outage constituted a contingency. Nine contingencies were included.

$$f(X_0) = \sum_{i=1}^{M} c_i P_{g\,i}$$

where, q = cost coefficient for generator i

Pg_i = real-power generation at the ith generator

$\tau_n(X_n\text{-}X_0) = HX_n\text{-}Xoll$, the euclidean norm

There are 10 sub-problems (base-case and 9 contingencies). Each sub-problem has 17 decision variables (generator real/reactive powers, bus voltage magnitudes, bus voltage angles [except slack]), and 47 constraints (12 equalities - power balance equations; 35 inequalities [loading limits {MVA} on 11 lines, bounds on real/reactive generation, bounds on bus-voltage magnitudes]). Thus (CCP2) has 170 decision-variables, and 470 constraints. A team of 10 voyagers (one for each sub-problem) and 10 probes (distributed over networked workstations) was used to solve the decomposed problem, and the results are tabulated in Table 1.

Four cases were studied.

Case 1: Only the base-case sub-problem was given an objective (f(xo) - only cost). The results show that the feasible regions are considerably

further apart - especially contingency-2 which would take more than a half-hour to correct.

Case 2: Contingency-2 was given sole consideration for the security-term in the objective. Cost was not considered. As is to be expected, the correction time for contingency-2 dropped considerably, while the times for the other contingencies increased.

Case 3: The security was "compromised" by adding a cost term to the objective. All contingencies were weighted equally. Compared to case 2, most of the correction times are significantly lower, except for contingency-2.

Case 4: This is similar to case 3 except that contingency-2 was weighted 10 times as much as the other contingencies. As a result, the correction time for this contingency reduced significantly.

Table 1
Results for a 6-bus, 11 -line, 3-generator system

Objective		Casei	Case 2	Case 3	Case 4
Base Case Cost ($)		19.32	-	20.03	18.65
Time (minutes) for correction for contin-gencies C1-C9	Cl	4.61	29.67	5.13	0.24
	C2	31.76	4.67	28.07	13.03
	C3	20.41	9.75	14.32	3.78
	C4	4.52	30.32	0.76	21.77
	C5	8.18	32.68	8.03	24.30
	05	11.90	29.41	5.97	21.20
	a	11.08	35.70	11.78	27.07
	CB	28.75	30.72	6.62	22.29
	09	3.24	30.42	0.82	22.02

3.2 **Global Optimization** Problems

The A-team approach was tested against a conventional approach (Random Multi-Start) for solving some difficult optimization problems [5]. The Random Multi-Start (R.M.S.) approach generates (random) points within a hypercube enclosing the feasible region. These points are used as launch points for the probes. The A-team consists of 6 copies of voyager and 3 copies of a probe distributed over 7 workstations (Dec 5000s connected by an ethemet). The SQP routine VF13 from the Harwell Library was used as the probe.

The optimization problems (He-6, Gr-10, Gr-15 and Gr-20) are described in the appendix. Both the methods were run till the global optimum was found. The results are averages over three trials. "Good" solutions are those for which the value of the objective function is within 30% of the value of the global minimum value. The function evaluation count is a combination of the function and gradient evaluations counts for both the objective and constraints. The formula used for the combination is:

function-evaluations +
(dimension x gradient-evaluations)

Table 2: Results for Optimization Problems

Problem	Method	no. of function evals required	no. of solutions found	no. of good solutions found
He-6	**A-Team**	**36,587**	13	**7**
	R.M.S.	**44,030**	10	3
Gr-10	**A-Team**	**217,963**	49	**47**
	R.M.S.	**659,853**	48	38
Gr-15	**A-Team**	**467,653**	49	49
	R.M.S.	**1,850,368**	45	**38**
Gr-20	**A-Team**	**1,454,397**	86	**85**
	R.M.S.	**5,916,456**	85	**71**

Two things are note-worthy:

- For all the problems, the A-team found more good solutions than the R.M.S. approach.
- The problems are tabulated as per increasing size and complexity (number of minima). He-6 is the least complex and Gr-20 the most. As the size increases, the A-team is significantly cheaper than the R.M.S. This is evidence that as the problem gets larger and more complex, the A-Team approach would be undoubtedly less expensive than the R.M.S. approach.

The results reflect the following advantages of the A-Team over R.M.S.:

1) Better (closeri launch-points

The voyager visits every basin & launches a launch-point therein which is, in general, closer to the nearest optimum than the corresponding random point generated by R.M.S. Hence the corresponding probe has to work less hard (and thus require lesser function evaluations) for the launch-points of the voyager.

2) Prevention of repeated re-determination of optima

R.M.S. wastes a lot of function evaluations in finding the same solution over again (several launch-points are generated leading to the same minima). A-Team prevents this through the use of fences.

3) Selection of launch-points

The heuristics for selecting launch-points eliminate lesser optima (a factor when there are lots of optima).

For He-6 factors (1) and (2) were important. For the other problems factors (1) and (3) were important.

4. SUMMARY

The original security planning problem is a computationally intractable global optimization problem. This paper has developed a new formulation, of the problem, consisting of several loosely-coupled sub-problems. A team of autonomous, asynchronous, concurrent and co-operative agents was described to solve this decomposed problem. Results were presented both for the security-planning problem, and for some other difficult global optimization problems.

5. REFERENCES

[1] Christie, R.D., Talukdar, S.N. and Nixon, J.C., CQR: A hybrid expert system for security assessment (IEEE Transactions on Power Systems, Vol. 5, No. 4, November 1990, pp. 1503-1509.)

[2] Talukdar, S.T., Pyo, S.S. and Mehrotra, R., Designing Algorithms & Assignments for Distributed Processing (Final Report for EPRI Research Project 1764-3, Nov., 1983.)

[3] Talukdar, S.N., Asynchronous Teams (Fourth Symposium on Expert Systems Application to Power Systems, Jan 4-8 1993, Melbourne, Australia.)

[4] Wood, A.J. and Wollenberg, B.F., Power Generation Operation and Control (John Wiley and Sons, 1984, pg 74, Fig. 4.1.)

[5] Torn, A. and Zilinskas, A., Global Optimization (Lecture Notes in Computer Science, No. 350, Springer-Verlag, 1989.)

APPENDIX

Description of test problems

Problem 1: He-6

$$\underset{x_1 \ldots x_6}{\text{Min}} \; -(25(x_i - 2)^2 + (x_2 - 2)^2 + (x_3 - 1)^2 + (x_4 - 4)^2 + (x_5 - 1)^2 + (x_6 - 4)^2$$

S.t. $x_1, x_2 \geq 0, 1 \leq x_3 \leq 5, 0 \leq x_4 \leq 6, 1 \leq x_5 \leq 5,$
$0 \leq x_6 \leq 10, 2 \leq x_1 + x_2 \leq 6, -x_1 + x_2 \leq 2,$
$x_i - 3x_2 \leq 2, 4 \leq (x_3 - 3)^2 + x_4, 4 \leq (x_5 - 3)^2 +$

Problem Statistics: 6 variables, 16 constraints

Solution
Global minimum at x* = (5, 1, 5, 0, 5, 10) with value -310

Problem 2: Gr-10

$$\underset{x_i}{\text{Min}} \sum_{i=1}^{10} x_i^2/4000 - \prod_{i=1}^{10} \cos(x_j/\sqrt{i}) + 1$$

s.t. $-600 \leq x_j \leq 600$, i = 1, 2,... 10

$$\sum_{i=1}^{10} x_i^2 \leq (1800)^2$$

Problem Statistics: 10 variables, 21 constraints

Solution
Global minimum at the origin with value zero.
There are several thousand local minima.

Problem 3: Gr-15

$$\underset{x_i}{\text{Min}} \sum_{i=1}^{15} x_j^2/80{,}000 - \prod_{i=1}^{15} \cos(x_j/\sqrt{i}) + 1$$

s.t. $-600 \leq x_j \leq 600$, i = 1, 2,... 15

$$\sum_{i=1}^{15} x_i^2 \leq (1800)^2$$

Problem Statistics: 15 variables, 31 constraints

Solution
Global minimum at the origin with value zero.
There are several thousand local minima.

Problem 4: Gr-20

$$\underset{x_j}{\text{Min}} \sum_{j=1}^{20} x_i^2/800{,}000 - \prod_{j=1}^{20} \cos(x_j/\sqrt{i}) + 1$$

s.t. $-600 \leq x_j \leq 600$, i = 1, 2,... 20

$$\sum_{i=1}^{20} x_i^2 \leq (1800)^2$$

Problem Statistics: 20 variables, 41 constraints

Solution
Global minimum at the origin with value zero.
There are several thousand local minima.