The Parallel Simulated Annealing-Based Reconfiguration Speedup Algorithm for the Real Time Distributed Control System Fault-Tolerance Providing

A.B. Klimenko
SFedU Acad. Kalyaev Scientific Research Institute of Multiprocessor Computer Systems
Taganrog, Russia
Anna_klimenko@mail.ru

V.V. Klimenko
SFedU Acad. Kalyaev Scientific Research Institute of Multiprocessor Computer Systems
Taganrog, Russia
V_klimenko@mail.ru

E.V.Melnik
OOO "Neirosetevye technologii
Taganrog, Russia
Evm17@mail.ru
"

Abstract **In this paper enhanced algorithm for the real time distributed control system fault-tolerance providing is presented. The algorithm is developed having the parallel simulated annealing technique as a base with the accent on the quenching temperature schemes. The combination of the parallel multistart technique and chosen temperature scheme allows to speed up to acquire the solution with the adequate quality. This provides system configurations in the acceptable time periods.**

***Index Terms*—fault-tolerant architecture, reliability, reconfiguration, information-control systems, fault-tolerant architecture model.**

I. Introduction

Modern automated process control system (APCS) are used in hazardous enterprises (chemical, petrochemical industry, hydroelectric and thermal power plants, nuclear power plants, etc.), or for the management of complex technological processes and systems (equipment with digital program management, aircrafts). APCS usually consists of a distributed control system (DCS) based on available industry data communication protocols, an automatic system of emergency protection (ESD) based on the telemetry subsystem, and a dispatching system (SCADA).

Typical technological network topology includes office segment of business process management, executive segment of the direct implementation of technological processes (for example, management of distributed control system (RTDCS)) and SCADA, where APCS operators can affect the progress of the process [1].

Modern RTDCS are used to control objects, that are unacceptable to be vulnerable[2-5], because it can lead to serious consequences. Nowadays the redundancy is the widespread solution of the RTDCS fault-tolerance problem. Also the distributed dispatching principles and computational unit software unification are proposed[6]. These features allow to reconfigure the RTDCS dynamically avoiding the total system failure.

The general scheme of such a system is presented below on fig. 1.

There is a set of computational elements (CEs), which runs some tasks according the predefined schedule. Each CE in the set is represented by the software agent which can be named "Local dispatcher"(LD). With the assumption of fully interconnected CEs, we have peer-to-peer multiagent system. All agents exchange status report messages.

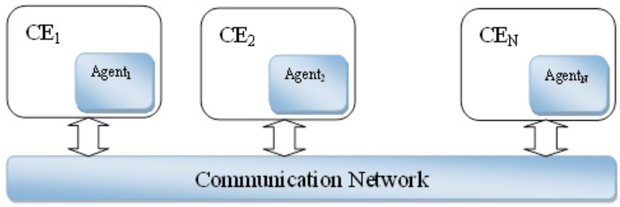

Figure 1: General scheme of RTDCS with the distributed dispatching

Generally, on system startup, each LD adopts the list of possible system configurations, where every identifier of faulted CE corresponds with the system configuration unambiguity. Hence, the reconfiguration process is fast and consists of adopting of the corresponding configuration. But if there is no corresponding configuration in the list (list has been corrupted, for instance), the multiagent system has to build a configuration «on the fly». The multiagent system must solve the problem cooperatively to get the solution as fast as possible.

More presizely, the situation described above can be presented as follows:

1. LDs exchanges status report messages.
2. In case of one or more LDs failure, LDs check the configuration lists. If there is corresponding configuration among the agent community, it is taken to be accomplished. If

there no such configuration, LD solves the resource allocation problem and sends the first viable result to CN. The best solution is chosen cooperatively to be accomplished.

The crucial aspect of the reconfiguration problem solution is the performance. Simulated annealing techniques are used for the np-hard problems solutions, but seemed to be too slow for the RTDCSs. The “quenching” temperature schemes allow to improve the computational process performance, but increase the probability to stuck in local minimas[7,8]. Historically, the multistart method improves the quality of the solutions, so the parallel multistart[9] does the same in the parallel manner.

II. RTDCS CONFIGURATION FORMING PROBLEM FORMALIZATION

Let the initial formal expressions be the following:

1) acyclic oriented graph, describing the control tasks:

$G=(H,X,W)$, where (1)

$H=\{j\}=\{1,\dots,n\}$ – the numbers of graph nodes,

$X=\{x_j\}$ – subtasks complexities,

W- set of graph edges, which describe the subtasks interconnections. Each edge is weighted by the w_{kl}, which is the size of data transitions between k and l subtasks.

2. Heterogeneous set of the CEs $M=\{m_i\}, i=1,\dots k$,

$m_i=<i, prod_i, v_i, s_i>$, where (2)

i – CE’s number,

$prod_i$ – CE’s productivity,

s_i – CE’s cost;

v_i – the speed of data transition through the communication network.

3) Planned moment of subtasks completion T_{plan}.

4) Actual time of subtasks completion T_c.

So, the formalization of the problem can be presented in the following way: $S = \underset{K}{MIN}\{\sum_i s_i\}$, $T_c \le T^{plan}$,

where

$K=\{k_1,\dots k_q\}$ – the variants of system structure

$S=\{s_1,\dots s_q\}$ – the summarized cost of the system.

The problem formalized is np-hard due to the necessity of scheduling forming.

III. RTDCS RECONFIGURING ALGORITHM: PARALLEL SIMULATED ANNEALING ADAPTATION

Let emphasize some key moments for the further algorithm synthesis:

- the objective function is the summarized cost of the system;
- for the next move evaluation we’ll use the Barskiy’s scheduling forming algorithm[10], with polynomial computaitional complexity.

Simulated Annealing (SA) is said to be the oldest among the metaheuristics and surely one of the first algorithms that had an explicit strategy to escape from local minima. In order to avoid being trapped in local minima, the fundamental idea is to allow moves to solutions with objective function values that are worse than the objective function value of the current solution. Such a move is often called an uphill move. At each iteration a solution s' is randomly chosen. If s' is better than s (i.e., has a lower objective function value), then s' is accepted as new current solution. Otherwise, s' is accepted with a probability which is a function of a temperature parameter T_k and $f(s') - f(s)$.

The “quenching” temperature schemes are used for the speedup the simulated annealing algorithms. Within this scheme, there is no guarantee to reach global minima, but satisfactory solutions can be produced with the adequate periods. Temperature change expression of the “quenching” simulated annealing is the following: $T_{k+1} = cT_k$, where

T_k – temperature on the move k,

T_{k+1} – temperature on the move $k+1$,

c – quenching ratio.

Simulated annealing is a serial search method, and it is the source of some difficulties of its parallelization. Nevertheless, some parallelization models were designed: parallel multistart (synchronous and asynchronous), parallel moves and move acceleration [11,12]. From the RTDCSs with the distributed dispatching point of view, the most suitable method is asynchronous parallel multistart (fig.2) in the reason of general independency of the search processes.

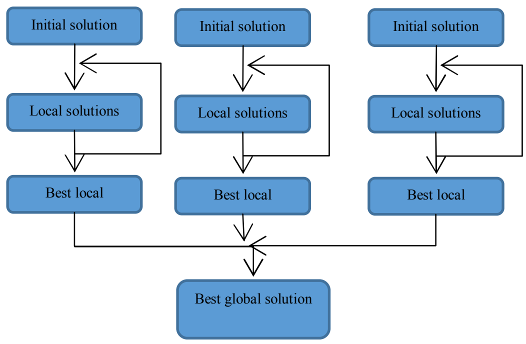

Figure 2: Parallel asynchronous multistart scheme

Let us look at the multi-agent simulated annealing adaptation. The temperature decreasing law will be according the “quenching” temperature scheme.

1. CE’s agent creates the list of tasks for the further assignment. Objective function value is shaped with the random initial point; the initial temperature is set too.

2. If there any „calculations end“ signs in the communication network, proceed step 7.

3. Generate a new solution.

4. Make a decision about the quality of the generated solution. If the decision is satisfactory, proceed step 7.

5. Check the constrains. If the decision is satisfactory, proceed step 7.

6. Make the temperature correction according to the following equation $T_{k+1} = cT_k$. Proceed step 2.

7. Put into the CN cost function value and the task-computational elements schedule.

8. Choose the best schedule from the available in the communication network.

9. Accept the decision for the implementation.

Let's take a look at the multi-agent simulated annealing adaptation.

IV. EXPERIMENTAL RESULTS

Within the paper presented, we consider to evaluate our algorithm in the following ways:

-to test the effectiveness of the quenching temperature scheme in the aspect of the best solution within the smaller computational time;

-to test the solution quality in dependency on the number of CEs.

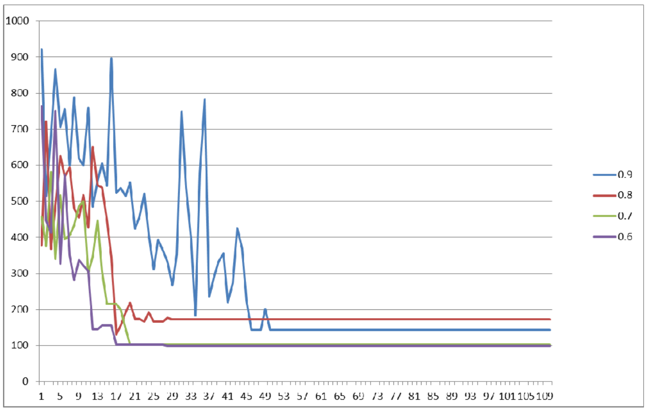

Figure 3: Simulated annealing: quenching ratios testing.

On the fig. 3, the algorithm convergence speed is presented for the 20 interconnected tasks with randomized complexities and 10 CEs with randomly produced productivity. The OX axe is marked with the number of objective function calls, the OY axe is marked with the objective function value (the total system cost in our case). The graphics allow to see that the fast temperature change can bring to the satisfactory result for the adequate time period (<20 objective function calls). So, if we don't search for the global minima, perhaps good local minima or the solution on the slop would be enough.

In addition, we compared the convergence results of the "quenching" simulated annealing with the simple genetic algorithm with one-point crossover and tournament selection. The size of population is 50 and 100.

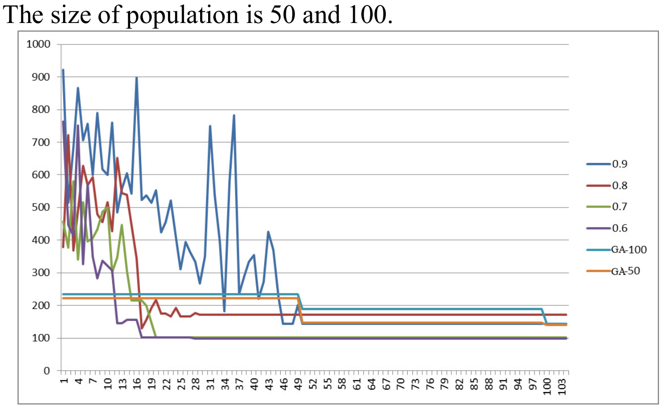

Figure 4: Gas and SA quenching schemes comparison.

The results of comparison allow to resume that simulated annealing with quenching temperature scheme can work faster than GA and allow to get the solution in a shorter time.

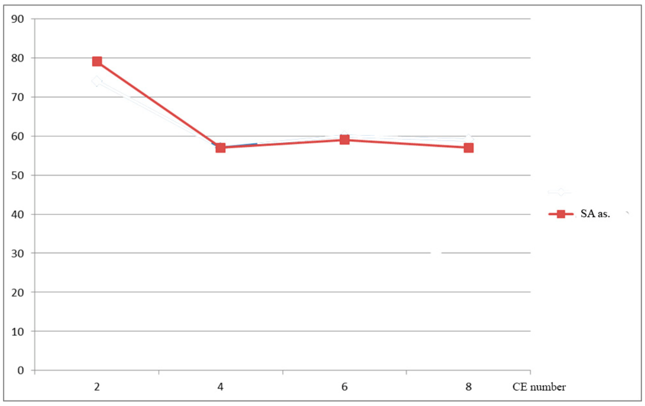

Figure 5: Solution quality evaluation in dependency on the CE number.

The next test of the algorithm is to evaluate the quality of solutions with the growth of the CE number. On the picture above, we tested our asynchronous adaptation on the 2, 4, 6 and 8 calculation threads. Obviously, with the increasing of the CEs number the quality of the solution is becoming better.

V. CONCLUSIONS

The main objective of the paper presented above was to evaluate the perspective of the parallel simulated annealing with the quenching temperature scheme for the RTDCS's reconfiguring. We synthesized parallel multistart simulated annealing adaptation for the fastened search procedure. The quenching temperature scheme with the heuristic scheduling forming were promising in the time consuming aspect.

VI. REFERENCES

[1] Overview of Cyber Vulnerabilities: Understanding Control System Cyber Vulnerabilities // https://ics-cert.us-cert.gov/content/overview-cyber-vulnerabilities#under

[2] Cyber Threat Source Descriptions // https://ics-cert.us-cert.gov/content/cyber-threat-source-descriptions

[3] Overview of Cyber Vulnerabilities: Access to the Control System LAN // https://ics-cert.us-cert.gov/content/overview-cyber-vulnerabilities#access

[4] Overview of Cyber Vulnerabilities: Control of the Process // https://ics-cert.us-cert.gov/content/overview-cyber-vulnerabilities #control

[5] The Cyber Security Evaluation Tool // https://ics-cert.us-cert.gov/Assessments

[6] Kalyev I. A., Melnik E.V.Decentralized systems of computer control. Rostov-on-don, SSC RAN publish, 2011. 196p.

[7] Kirkpatrick, S.; Gelatt, C. D.; Vecchi, M. P. (1983). "Optimization by SimulatedAnnealing".Science220 (4598): 671–680

[8] L.Ingber, Simulated annealing: practice versus theory.1993. (http://citeseer.uark.edu:8080/citeseerx/viewdoc/summary?doi=10.1.1.15.1046)

[9] S.Y. Lee and K.G. Lee. Synchronous and asynchronous parallel simulated annealing with multiple markov chains. IEEE Transactions on Parallel and Distributed Systems, 7:993-1008, 1996.

[10] Barskiy A.B. Parallel processes in computation systems. Moscow, Radio and communications, 1990

[11] S.A. Kravitz and R.A. Rutenbar. Placement by simulated annealing on a multiprocessor. IEEE Trans. in Computer Aided Design, 6:534-549, 1987.,

[12] M. Miki, T. Hiroyasu, and M. Kasai. Application of the temperature parallel simulated annealing to continuous optimization problems. IPSL Transaction, 41 : 1607-16 16,2000.