CASOS

CarnegieMellon

A Network Optimization Approach for Improving Organizational Design[1]

CASOS Technical Report

Kathleen M. Carley, Natalia Y. Kamneva

January 2004

CMU-ISRI-04-102

Carnegie Mellon University

School of Computer Science

ISRI - Institute for Software Research International

CASOS - Center for Computational Analysis of Social and Organizational Systems

Abstract

Organizations are frequently designed and redesigned, often in efforts to improve performance or meet various managerial goals for coordination and communication. Such design is often done through the review of a few of options and the use of managerial and possibly personnel insight into how the new design might work. In contrast, we provide a systematic optimization based approach. In this approach, the user can pick one or more Dynamic Network Analysis (DNA) metrics and then use one or more of the available optimizers to find a design that more closely meets this ideal. The optimizer utilizes heuristic based optimization procedures to generate an optimized organizational design given a particular mission. DNA metrics, such as Communication Congruence, Resource Congruence, Cognitive Load, and Actual Workload, serve to define criteria. The Optimizer can perform multi-criteria optimization in order to improve several metrics simultaneously. Two optimization methods can be used - Monte Carlo and Simulated Annealing, both of which are statistical methods of finding a global optimum. DNA metrics used in the optimizations are computed by ORA. This report describes this optimizer.

University Libraries
Carneeie Mellon University
Pittsburgh, PA 15213-3890

Table of Contents

List of Tables

List of Figures

1. Introduction

Organizations are often grow to have a particular structure or design, that, when examined can be seen to being fraught with defects. In contrast, technologies and various products are often engineered so that they exhibit a design that meets various goals. Many products are even optimized to meet some set of specifications. The same techniques, however, can be applied to organizations. We can ask, how should we design this organization, this group, this team so that it is "optimal" given some set of design criteria.

Today, information processing, communication, and knowledge management are keys to effective organizational performance and adaptability. Changes in computational power, telecommunications, and information processing are affecting when, where and how work is done [1, 2]. Further changes in agriculture, manufacturing, transportation and technology are leading to the emergence of an increasingly mobile population and knowledge intensive organizations. New organizational designs are emerging such as network organizations [3,4] and virtual organizations [5]. In these new organizations, even though information processing is key [6], communication is not constrained to be vertical [7]. Rattier, the network of connections within and among organizations acts to constrain and enable the flow of goods, services, agents and information. The result is an environment in which the act of organizational design becomes a strategic exercise in establishing and managing these relations [8].

What are these relations? How can we, given the set of possible relations, find the optimal design? What are the criteria for determining whether a design is good? What are the appropriate optimization algorithms? In this report, we provide a first answer to these questions.

What Are the Relations?

A variety of networks exist within and among organizations. We can define a meta-matrix [9] as the networks connecting the four key corporate entities - agents, knowledge, resources, and tasks (see Table 1). Various aspects of organizations can be characterized in terms of these networks. For example, structure (such as the authority structure or the communication structure) is defined in terms of the interaction network connecting people to people. Culture can be defined in terms of the knowledge network - the connections.of people to knowledge. And so on.

The individual cells in this meta-matrix define items that can be manipulated by the manager. The goal from a design perspective is to alter the elements of Table 1 to achieve a design that meets a set of criteria Clearly there are different constraints and costs on manipulating various aspects of this meta-matrix... In general these cells can be manipulated by adding or dropping nodes and adding or dropping relations. Logically, the organization can be changed by adding or dropping nodes or relations. A node can be, given Table 1, a person, knowledge, resource, or task. A relation can be the connection between two nodes. Further, unlike nodes, we can talk of change in the strength of a connection. A number of key processes in covert networks affect these types of changes. Key processes affecting node change include: recruitment; the removal (death, isolation, etc.) of a person; change in mission (and so the addition or deletion of tasks); change in technology (and so the addition or deletion of tasks and resources); the consumption of resources; and the purchase/creation of resources. Key processes affecting the change in relations are re-assignment of personnel, training, co-work assignments, and evolution of friendships/communication structure.

Table 1: Meta-Matrix for Organizational Design

	People	Knowledge	Resources	Tasks
People	Interaction Network	Knowledge Network	Resource Network	Assignment Network
Knowledge		Information Network	Resource Skill Needs Network	Task Skill Needs Network
Resources			Substitutes and Coordinated Resources Network	Task Resource Needs Network
Tasks				Task Precedence Network

In most organizations, due to missing and capital outlays (fixed investments) changes to tasks and resources is harder than changes to people and knowledge at least in the short run. Thus, assuming that the set of tasks is more or less set by the organizational mission and the extant technology, then things involving adding/dropping tasks or connections among tasks will be difficult to do in the short run. In contrast, things involving people are easier to do in the short run. We can thus think of row 1 of Table 1 as representing that part of the organization that can be changed, manipulated, altered by the manager fairly quickly. The rest of the organization we can think of as the more fixed, less malleable, aspect of the current design. This characterization constrains the optimization problem to manipulating the top four cells - the interaction, knowledge, resource, and assignment networks to meet needs defined in part by the portions of the current design that are "fixed".

A great deal of research demonstrates that for humans, the networks that people operate on, and in, serve to constrain and enable further action and affect the efficiencies of such actions [10]. Similarly, for artificial agents, being able to traverse the digitized version of these networks enables machine comprehension [11]. For example, WebBots that serve as personal shoppers are more intelligent if they are more able to navigate through the links between sites on the web.

Hence a change in any one of the four networks in which people are involved can potentially result in a cascade of changes in the others. For example, when individuals learn something new (by interacting with someone in their interaction network) that evokes a change in the interaction network [12]. As another example, when new personnel are hired they may bring new knowledge with them. As current personnel leave, the available knowledge may be depleted. From an optimization perspective the goal would be to find the set of changes that results in a cascade that meets the organizational goals.

Managing these changes is the key to knowledge management. Information technology has the potential to affect this meta-network in several ways. First, it can affect the number and types of nodes in these networks; i.e. with the advent of new technology comes new agents, new knowledge, and new connections among knowledge. Second, information technology has the potential to alter the way changes occur and their impact. For example, some suggest that

holding data in databases, and knowledge systems like Lotus Notes provides organizations with the means to decouple personnel turnover and change in the knowledge network.

By identifying the mission and technology as the constrained portions or the relatively fixed components of the extant system, at least in the short run, we open the possibility to locating the optimal form or structure of the rest of the system. We define the organizational design problem in terms of the meta-matrix that can be varied in the short run - the interaction network, the knowledge network, the resource network, and the assignment network. The system is optimized if the ties in this network are arranged such that they minimize those vulnerabilities of concern to the manager.

What are these vulnerabilities? How can we define the set of them? By defining the organizational design in terms of a set of networks that open the possibility of using all networks (both social and dynamic network measures) as indicators of potential vulnerabilities. Further, we know from the past decade of work on organizational design, that many of these measures are directly related to encourage adaptation whereas others encourage high performance. For example, previous work indicated that high performance and adaptive systems tended to exhibit a high level of congruence, or match, between what resources were needed for a task and the availability of those resources (resource congruence) and in who needed to communicate in order to do the task and who actually communicated [13]. Further, organizations typically exhibit better performance and have fewer problems with personnel if workload is evenly distributed. Using heuristic based optimization tools, such as simulated annealing and Monte Carlo techniques, we have developed a series of procedures, that given meta-matrix data on an organization locates the organizational design that optimizes one or more of these criteria.

There are two ways in which the optimization code can be used. First, it can be used to assess the extent to which the organization as a whole is in trouble. For example, if the current design is far from optimal it may not be worth destabilizing at all. Since destabilization involves the removal of critical nodes, the comparison of the relative difference from the optimum of the "destabilized" organization and original provides an indicator of the potential relative impact of the destabilization. Secondly, this tool can be used by a manager to locate possible new designs.

2.   Concept of Optimization: Our Case

The optimizer utilizes heuristic based optimization procedures to generate an optimized organizational design given a particular mission. Dynamic Network Analysis (DNA) metrics, such as Communication Congruence, Resource Congruence, Cognitive Load, and Actual Workload, can be used individually or in combination as objective functions to be minimized or maximized. By combining several DNA measures, either via sum or product, multi-criteria optimizations can be performed.

The space over which the DNA metrics are defined is not the N-dimensional space of N continuous parameters. In our case, N is either the number of nodes in the meta-matrix, or the total number of edges, and our sample space is a discrete with size proportional to 2^N. Because the set is discrete, we are deprived of any notion of "continuing downhill in a favorable direction." The concept of "direction" does not have any meaning in the configuration space, and therefore we cannot use gradient or pseudo gradient methods to optimize our objective functions. On the other hand, the sample space is exponentially large, and so it cannot be explored

exhaustively. The optimization methods used by Optimizer are therefore statistical sampling methods: Monte Carlo and Simulated Annealing.

3. **Input for Optimization**

The Optimizer takes as input an organization represented as a meta-matrix. In the short run we assume that the number of entities (people, resources, knowledge, and tasks) is fixed. It is also assumed that for the purposes of optimization, some matrices in the meta-matrix are constant, and some are variable (see Table 2). So, we define the organizational design as the set of cells in the meta-matrix that can be varied in the short run - the social network, the capabilities network, the assignment network, and knowledge network. The system is optimized if the ties in those networks are arranged such that they minimize vulnerabilities. We define a system to have the optimal organizational configuration or design if vulnerabilities due to one or more of the following are minimized: distribution of resources, distribution of communication ties, and workload.

Previous work indicated that high performance and adaptive systems tended to exhibit a high level of congruence, or match, between what resources were needed for a task and the availability of those resources (resource congruence) and in who needed to communicate in order to do the task and who actually communicated. Furthermore, organizations typically exhibit better performance and have fewer problems with personnel if workload is evenly distributed [13].

In the case when the original input data does not have a required matrix of the meta-matrix, we can always create it as a random matrix at the beginning of the optimization process.

Table 2: Meta-matrix shown with fixed and variable components.

	People	**Knowledge**	**Resources**	**Tasks**
People	variable component	variable component	variable component	Variable component
Knowledge			fixed component	Fixed component
Resources				Fixed component
Tasks				Fixed component

4. **Output of Optimization**

As previously mentioned, DNA metrics serve to define criteria. As a single criterion of the optimization we used the DNA metrics Communication Congruence, Resource Congruence, Cognitive Load, Actual Workload, Communication, and Personnel Cost. For the definitions of these measures, see the Organization Risk Analyzer (ORA) Technical Report. These measures were chosen because previous work suggests that to improve performance of the organization we

should maximize Congruence, both Communication and Resource, and minimize variation in Actual Workload. On the other hand, to improve adaptability we should minimize the variation in Cognitive Load. Hence, optimizing a meta-matrix for performance or for adaptability requires optimization with criteria created as a sum or a product of the DNA metrics.

Different DNA metrics require different sub matrices to be varied during the optimization process. Resource Congruence requires assignment and capabilities networks, Communication Congruence, Communication, and Cognitive Load additionally require social network, Actual Workload requires assignment and knowledge networks, and Personnel Cost requires all 4 variable matrices to be completely optimized. However due to unwillingness of the user to change all variable matrices we always can consider only some of them as variable. On the other hand if we do not originally have any sub matrices, we always can consider them as variable and simulate them during the process of the optimization. Finally after the optimization process we will get all variable sub matrices close to their optimal meaning.

5. Optimization Methods

While the classical approach would obviously not be applicable to our problem, statistical heuristic techniques seem, intuitively, to be appropriate for our purpose. The literature survey demonstrated that each technique had its strengths and weaknesses. It also demonstrated that the performance of each algorithm would be heavily dependent on the nature of the problem itself and the heuristics that we used. We chose two statistical optimization methods: Monte Carlo and Simulated Annealing to use in the Optimizer.

5.1. Monte Carlo Method

The Monte Carlo method randomly samples the variable sub matrices of our meta-matrix.

We randomly generate all cells in a meta-matrix with uniformly distributed random densities of sub-matrices. At each sample point, the objective function is evaluated. After N experiments we take as an approximation to the global optimum the sampled meta-matrix that yielded the highest objective function.

The advantages of Monte Carlo method for solving our problem are:

It provides a broad sampling of the parameter space, and gives the possibility of finding the global optimum.

It allows random samples to be generated subject to structural constraints, such as each row in a sub-matrix having at least one non-zero element.

It allows simulating of the sub-matrices with fixed or randomly distributed densities.

Its disadvantages are that it can be slow (if many experiments are selected), and that because of the random, discrete nature of the search, the global optimum can easily be missed (if not enough experiments are selected).

5.2. Simulated Annealing Method

The rough idea of simulated annealing is that it first picks a random move. If the move improves the objective function, then the algorithm accepts the move. Otherwise, the algorithm makes the move with some probability less than 1:

Formula 1 $$p = \exp\left[-(E2-E1)/kT\right];$$

The probability decreases exponentially with the "badness" of the move - the amount $(E2 - E1)$ by which the evaluation is worsened.

A second parameter T is also used to determine the probability. At higher values of T, "bad" moves are more likely to be allowed. As T tends to zero, they become more and more unlikely, until the algorithm behaves more or less like local search. The schedule input determines the value of T as a function of how many cycles already have been completed [14].

The algorithm was developed from an explicit analogy with annealing – the process of gradually cooling a liquid until it freezes. The objective function corresponds to the total energy of the atoms in the material, and parameter T corresponds to the temperature. The schedule determines the rate at which the temperature is lowered. Individual moves in the state correspond to random fluctuations due to thermal noise. One can prove that if the temperature is lowered sufficiently slowly, the material will attain a lowest-energy (perfectly ordered) configuration. This corresponds to the statement that if schedule lowers T slowly enough, the algorithm will find a global optimum. Simulated Annealing was first used extensively to solve VLSI layout problems in the early 1980s [14, 15]. Since that, it has been used in Operations Research to successfully solve a large number of optimization problems such as the Traveling Salesman problem and various scheduling problems [14].

The advantages of Simulated Annealing method are:

It is not "greedy," in the sense that it is not easily fooled by the quick payoff achieved by falling into unfavorable local minima.

If it doesn't find the absolutely best solution, it often converges to a solution that is close to the true minimum solution.

It takes less time than Monte Carlo method to get a comparable solution.

The disadvantages of Simulated Annealing method are:

It does not easily allow the logical constraints on the solution: for example, keeping at least one 1 in every row of sub-matrices.

It does not allow simulating of the sub-matrices with fixed densities.

One of the difficulties in using Simulated Annealing is that it becomes very difficult to choose the rates of cooling and the initial temperatures for the system that is being optimized.

This occurs primarily because of the absence of any rules for selecting them. The selection of these parameters depends on heuristics and varies with the system that is being optimized.

6. Performance Information

To test the optimizer we used network data from the U.S. embassy bombing in Tanzania. The network is small, with 16 agents, 4 knowledge, 4 resources, and 5 tasks.

Some results of the optimization using the optimizer were also presented in [17].

We optimized this data using both methods: Monte Carlo and Simulated Annealing. For Monte Carlo, we used the default case with the number of experiments N = 1,000,000 and the user specified case with a significantly smaller number of experiments N = 100,000. We also considered two different cases of optimization with a single measure to be optimized and with all four measures optimized using the sum criterion. The results of the optimization using Monte Carlo are presented in Table 3. All table results are from running the Optimizer on a 2.53 Ghz Intel Pentium IV processor running Windows XP.

For Simulated Annealing, we used the default case with optimization parameters T = 100 (original temperature) and T_factr = 0.99995 (coefficient regulated cooling schedule). The user specified case was T = 90 and Tfactr = 0.995 that required significantly smaller time in the optimization process. We also considered two different cases of optimization with a single measure to be optimized and with all four measures optimized using the sum criterion. The results of the optimization using Simulated Annealing method are presented in Table 4.

The time of optimization in minutes is presented in Tables 5 and 6 for Monte Carlo and Simulated Annealing, respectively. The experiments were done with the networks that contained 25, 50, 100, 200, 500, and 1000 nodes in total. During these experiments the networks with one variable sub-matrix, two variable sub-matrices, and three variable sub-matrices were optimized. Also the optimization was provided for one measure or for four measures combined using the sum criterion.

Table 3: Optimizatioii Results for Monte Carlo

Optimized Measure	Default N = 1,000,000			User Specified N = 100,000		
	Non-Optimized Value	Optimized Value	Optimized Using All 4 Measures	Non-Optimized Value	Optimized Value	Optimized Using All 4 Measures
Resource Congruence	0.4	0.9500	0.9500	0.4	0.9000	0.8500
Comm. Congruence	0.425	0.7333	0.5250	0.425	0.7000	0.5833
Std Dev in Cog Load	0.0824	0.0164	0.0516	0.0824	0.0186	0.0702
Std Dev in Work Load	0.2644	0.0000	0.1491	0.2644	0.0000	0.1596
Avg. Cog Load	0.1329	0.1290	0.1290	0.1329	0.1298	0.1319
Avg. Work Load	0.2292	0.0000	0.0833	0.2292	0.0000	0.1458

Table 4: Optimization Results for Simulated Annealing

Optimized Measure	Default T = 100; T.Jàctr = 0.99995			User Specified T = 90; 'T_factr = 0.995		
	Non-Optimized Value	Optimized Value	Optimized Using AU 4 Measures	Non-4Optimized Value	Optimized Value	Optimized Using AU 4 Measures
Resource Congruence	0.4	1.0000	1.0000	0.4	1.0000	1.0000
Comm. Congruence	0.425	1.0000	0.8833	0.425	0.8250	0.6292
Std Dev in Cog Load	0.0824	0.0042	0.0259	0.0824	0.0098	0.0352
Std Dev in Work Load	0.2644	0.0000	0.0000	0.2644	0.0000	0.0860
Avg. Cog Load	0.1329	0.2461	0.2057	0.1329	0.3026	0.2089
Avg. Work Load	0.2292	0.0000	0.0000	0.2292	0.1667	0.0833

Table 5: Time of Optimization in Minutes for Monte Carlo, N = 1,000,000.

Number ol nodes	One measure optimized			Four measures optimized		
	1 sub matrix	2 sub matrices	3 sub matrices	1 sub matrix	2 sub matrices	3 sub matrices
25	7	10	15	16	23	34
50	16	24	34	37	55	78
100	34	50	71	78	115	163
200	68	100	142	156	230	327
500	484	710	1008	1113	1633	2318
1000	1936	2840	4032	4452	6532	9274

Table 6: Time of Optimization in Minutes for Simulated Annealing, T = 100; t_factr = 0.9995.

Number of nodes	One	measure	optimized	Four	measures	Optimized
	1 sub matrix	2 sub matrices	3 sub matrices	1 sub matrix	2 sub matrices	3 sub matrices
25	0.15	0.18	0.20	0.38	0.52	0.52
50	0.56	0.73	1.64	0.76	1.67	1.66
100	2.18	2.47	5.03	4.48	5.18	7.11
200	9.05	9.64	25.8	29.53	28.57	40.01
500	61.00	82.37	100.8	323.48	345.67	387.70
1000	363.72	605.10	950.68	1440.85	1650.29	1847.92

Comparison of the optimization time for Monte Carlo and Simulated Annealing is also presented in Figure 1. The experiments have been done with the networks that contained number 25, 50, 100, 200, 500, and 1000 total nodes. During these experiments the networks with one variable sub-matrix, two variable sub-matrices, and three variable sub-matrices were optimized. Also the optimization was provided for one measure or for four measures combined using the sum criterion.

Figure 1: Dependency of Optimization Time for Simulated Annealing and Monte Carlo for 1 measure and 4 measures using the sum criterion
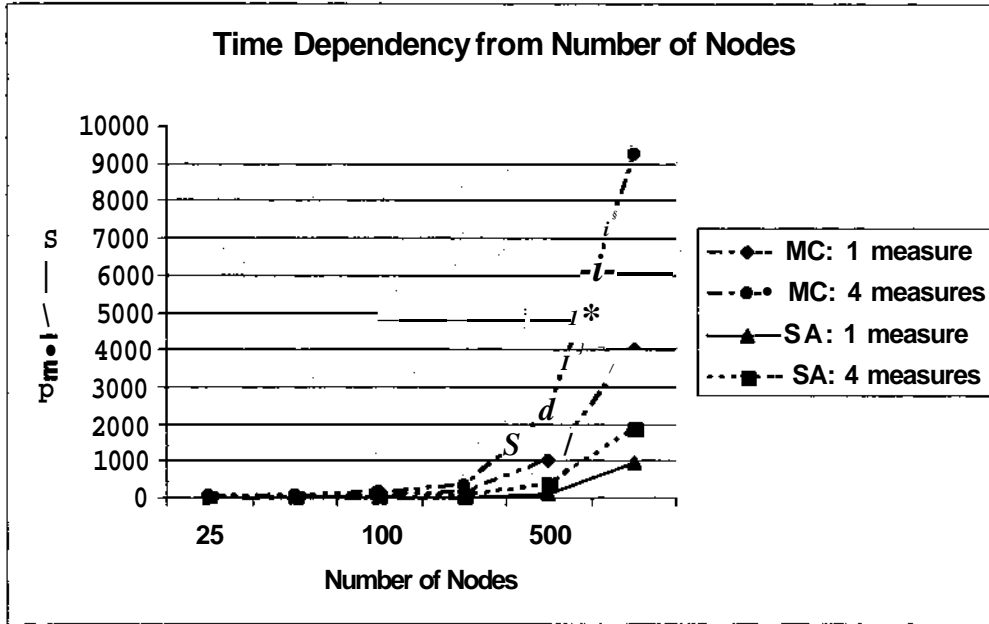

Table 7: Optimization by Simulated Annealing and Monte Carlo for Measures Communication and Personnel Cost

	Communication		Personnel Cost	
	Simulated Annealing	Monte Carlo	Simulated Annealing	Monte Carlo
Original Value	0.9958	0.9958	0.4640	0.4640
Optimized Value	0.0040	0.3000	0.0893	0.1518

Some additional optimization results are presented in Table 7. The table displays the results of the minimization of two additional measures of the Embassy Bombing Network - Communication and Personnel Cost. Both methods required approximately the same amount of time, but Simulated Annealing gives considerably better results that Monte Carlo method.
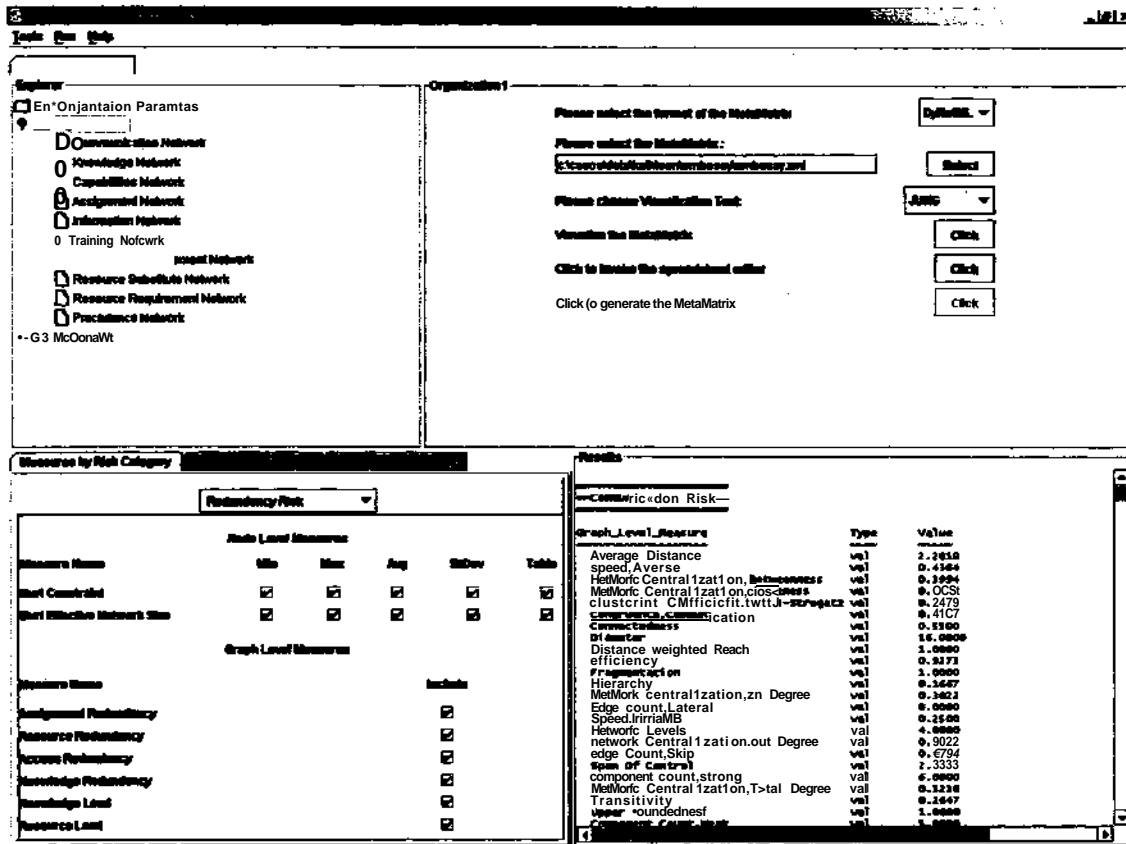
7. Two Different Versions of the Optimizer

The optimizer currently has two different versions. Both of them are integrated with netstatplus. The first version is also combined in the ORA interface. This version additionally includes the parser for parsing the output files written in xml format and coming from Java written interface to the C++ coded optimizer. This version takes all the parameters for the optimization from the ORA interface. It is simple for sophisticated and non sophisticated users to make deal with: to set, specify, and change methods, parameters, models, and datasets. It is also very usable for comparison of the original version of the dataset and corresponding metrics and the optimized version.

The second version of the optimizer is the so called no-GUI version. It makes possible for the setting of all the parameters directly to the optimizer. While this version does not allow the user to manipulate the parameters, it is much faster since it does not have to parse the xml files. This version is extremely useful for operating with huge datasets containing more than 500 - 1000 nodes when it becomes impossible to describe them in the xml format. It is also efficient for the simple and fast optimization tasks and for use with some other than netstaplus applications. In this version the user just prompts for the method of optimization, type of criterion, number and type of measures to be optimized.

8. Design of the User Interface

The Optimizer has a Java user interface combined with ORA interface (Figure 2) and described in detail in [18]. The Optimizer is invoked from the main menu and is contained within pop-up windows.

Figure 2 : ORA/Optimizer Interface

The user can choose the method of optimization: Monte Carlo or Simulated Annealing (Figure 3).

Figure 3: The Optimizer GUI: selection of the optimization method.

Optimizer

Choose Optimization Method:

⟨§⟩ Monte Carlo

O Simulated Aimeaino

| <- Previous | | Next-> |
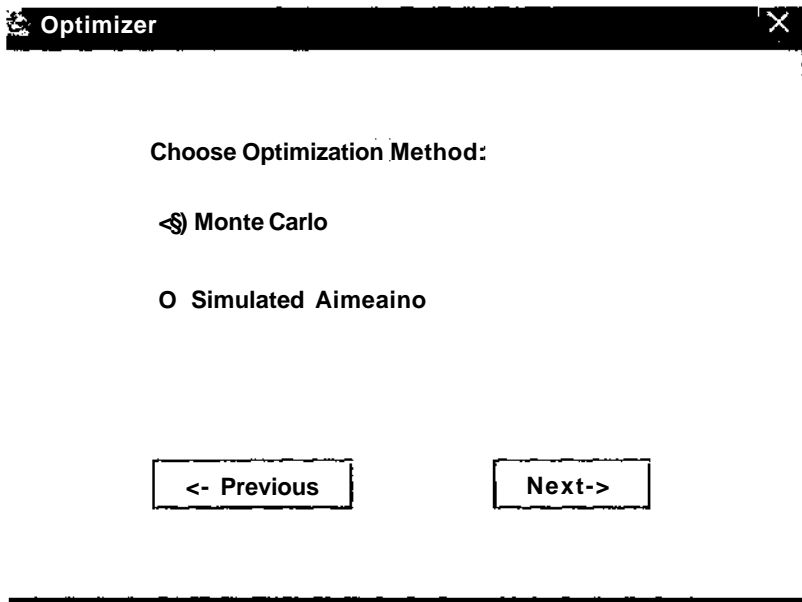
There are default versions of each method with default parameters of the optimization process (Figure 4) and also there is the option to change optimization parameters such as N and T (Figure 5).

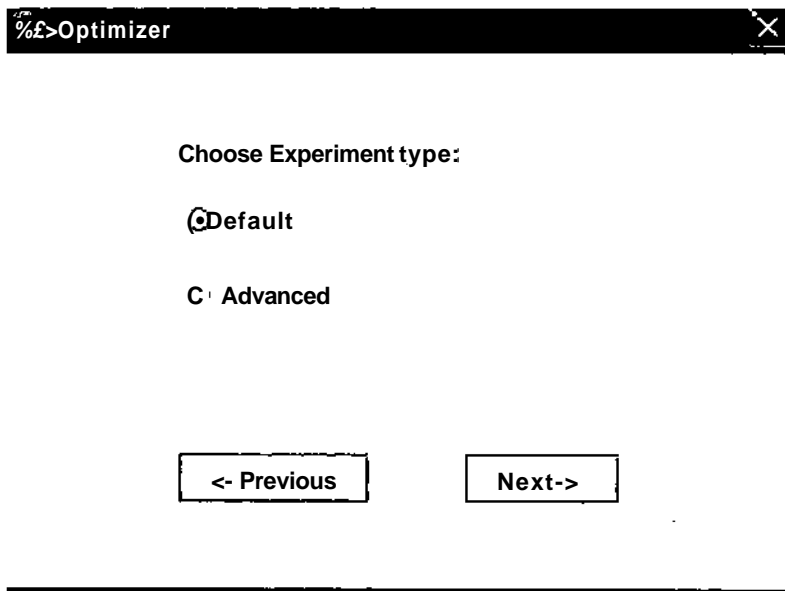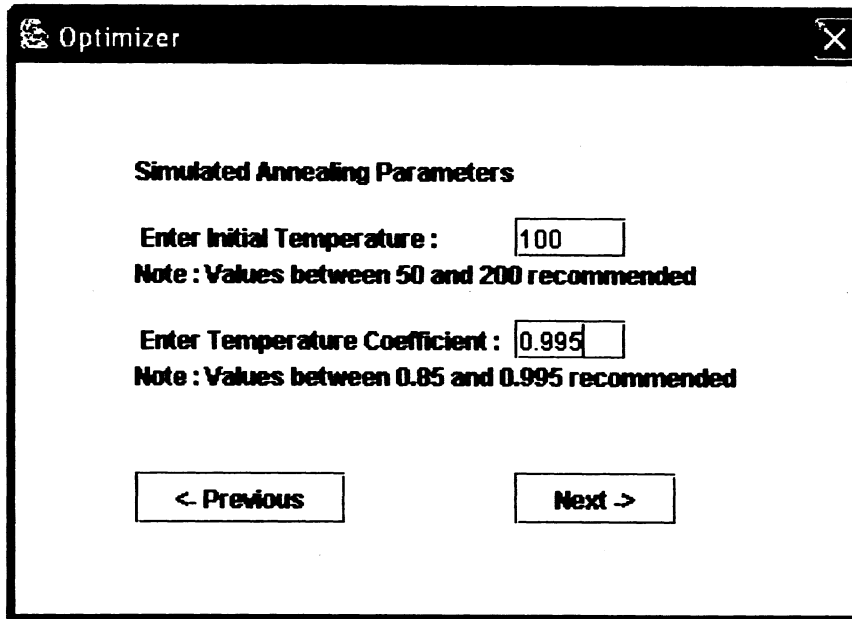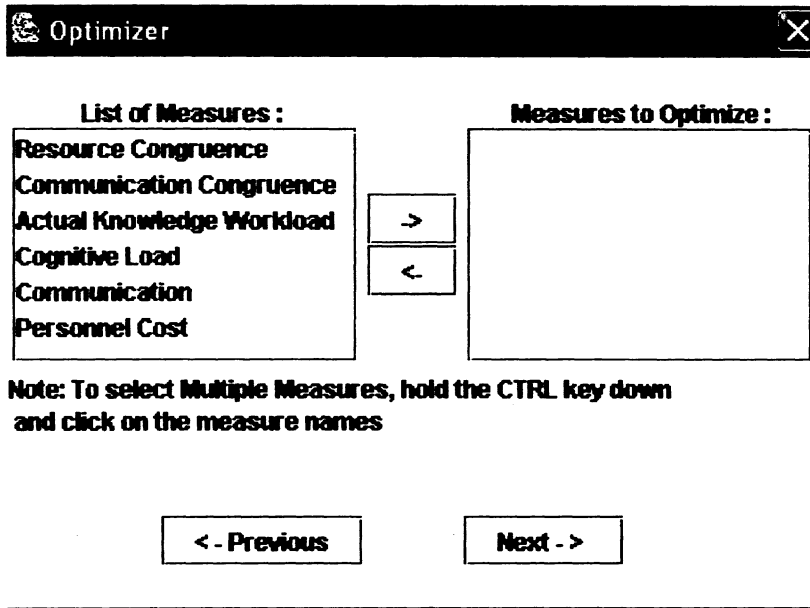Figure 4: The Optimizer GUI: selection of the experiment type.

%£>Optimizer

Choose Experiment type:

⊙Default

C Advanced

| <- Previous | | Next-> |

Figure 5: The Optimizer GUI: selection of the optimization parameters.

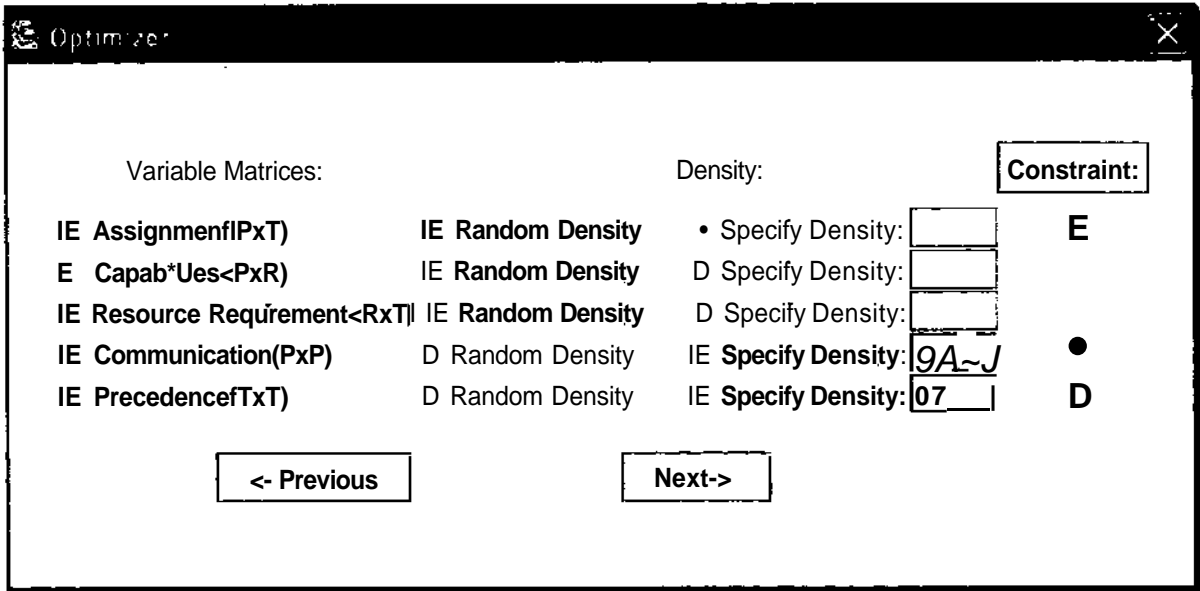The user can choose how many and which measures should be optimized (Figure 6).

Figure 6: The Optimizer GUI: selection of the optimized measures.


The user can choose which sub-matrices should be varied and which should be fixed during the optimization process (Figure 7). For the Monte Carlo method the user has the option to optimize with fixed or varied densities of sub-matrices (Figure 7).

Figure 7: The Optimizer GUI: selection of the variable matrices, specification of their densities and constraints.


Figure 7: The Optimizer GUI: selection of the variable matrices, specification of their densities and constraints.

Optimizer

Variable Matrices:		Density:	Constraint:
IE Assignmenf[PxT)	IE Random Density	• Specify Density: [ ]	E
E Capab*Ues<PxR)	IE Random Density	D Specify Density: [ ]	
IE Resource Requirement<RxT|	IE Random Density	D Specify Density: [ ]	
IE Communication(PxP)	D Random Density	IE Specify Density: 9A~J	●
IE PrecedencefTxT)	D Random Density	IE Specify Density: 07___|	D

<- Previous   Next->

The user can choose the optimization criteria: single, sum, or product (Figure 8).

Figure 8: The Optimizer GUI: selection of the optimization criterion.

Optimizer

Choose Optimization Criterion:

◉ **Sum**

O **Product**

<-Previous   Mext->
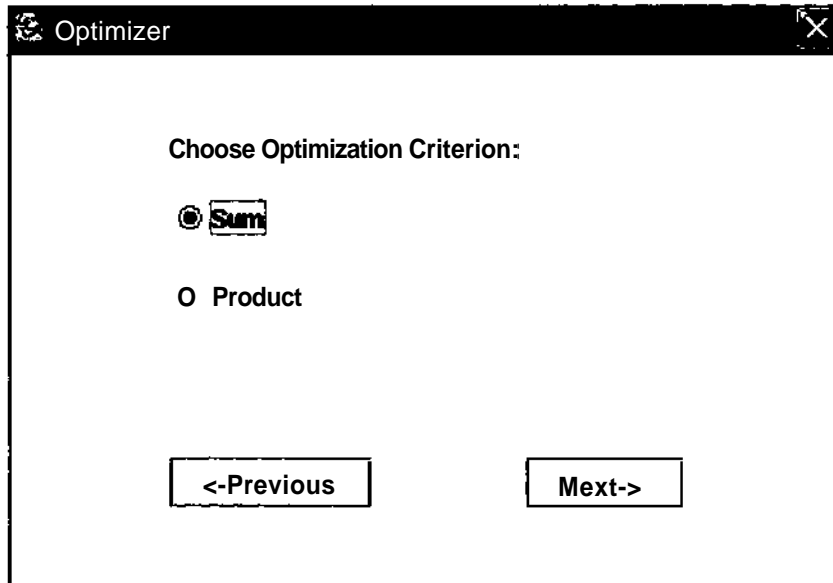
The user can specify the brief or verbose forms of text file output (Figure 9).

Figure 9: The Optimizer GUI: selection of the output format
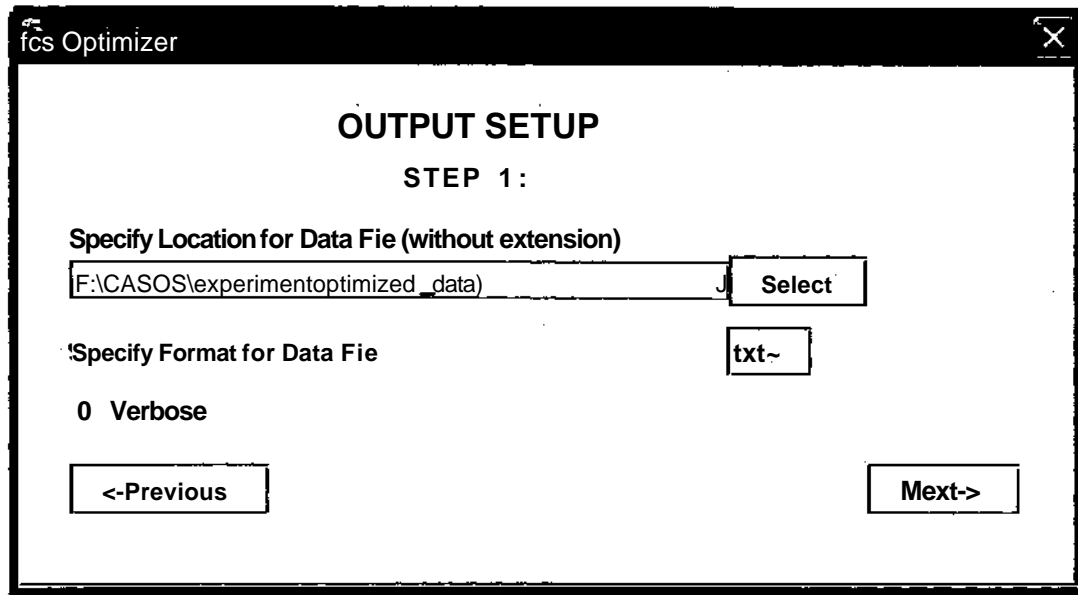
9. Limitations and Future Extensions

The main limitation of this work is that we operated with a pretty small data set. As you can see from Table 5 and 6 the large data sets with a number of nodes exceeded SOO can hardly be optimized with current algorithms and hardware. So we are going to use some modifications of currents algorithms, combination of Monte Carlo and Simulated Annealing to decrease run time, for example. It is also possible to use constraints on some rows and columns of sub matrices to decrease the size of the variable networks. It will automatically lead to decreasing the optimization time.

We also plan to increase the number of metrics being optimized. Practically the optimizer can be used with any DNA metric. It is possible to consider some other optimization criteria besides sum and product. For example, it might be minimax criterion when on every step of optimization we try to optimize the worst criterion. This strategy eventually leads to improving all metrics.

We also consider the possibility to explore the linkage of optimizer to scheduler using the optimized designs as constraints on scheduling.

10. System Requirements

The Optimizer runs as part of ORA, which is freely available from the CASOS website. The front end of ORA is written in Java, and the back end in a C++ network analysis library called NetStatPlus. The Optimizer is written in C++ and currently runs on Windows XP using an Intel processor, and the code has been actively ported and tested on other platforms.

References

[I] Di Martino, V. & L. Wirth, 1990. "Telework: A New Way of Working & Living." International Labour Review, 129(5): 529-554.

[2] Sproull, L., & S. Kiesler, 1991. Connections: New Ways of Working in the Networked Organization. Cambridge, MA: MIT Press.

[3] Nohria, Nitin & Robert G. Eccles (Eds.), 1992. Networks and Organizations: Structure, Form, and Action, Boston, MA : Harvard Business School Press.

[4] Miles, R. E. & C.C. Snow, 1995. The New Network Firm: A Spherical Structure Built on a Human Investment Philosophy. Organizational Dynamics, 23, 5-18.

[5] Lipnack, Jessica & Jeffrey Stamps, 1997. Virtual Teams: Reaching Across Space, Time, and Organizations with Technology. New York: John Wiley & Sons.

[6] Tushman, M.L. & D. Nadler, 1978, Information Processing as an Integrating Concept in Organizational Design. Academy of Management Review, 3: 613-624.

[7] Contractor, N. & E. Eisenberg, 1990. Communication Networks and New Media in Organizations. In Organizations and Communication Technology, J. Fulk and C. Steinfield, eds. Newbury Park: Sage Publications.

[8] Burton, Richard M. & Borge Obel, 1998. Strategic Organizational Design: Developing Theory for Application, Kluwer Academic Publishers.

[9] Carley, Kathleen M., 2003, "Dynamic Network Analysis" in Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers, Ronald Breiger, Kathleen Carley, and Philippa Pattison, *(Eds.)* Committee on Human Factors, National Research Council, National Research Council. Pp. 133-145.

[10] Burt, Ronald S., 1992. Structural Holes: the social structure of competition. Cambridge, MA: Harvard University Press.

[II] Bookman, Lawrence A. 1994. Trajectories through Knowledge Space: a Dynamic Framework for Machine Comprehension. Boston: Kluwer Academic Publishers.

[12] Carley, Kathleen M., 1991. "A Theory of Group Stability." American Sociological Review, 56(3): 331-354.

[13] Carley, Kathleen M, 2002, "Inhibiting Adaptation" In Proceedings of the 2002 Command and Control Research and Technology Symposium. Conference held in Naval Postgraduate School, Monterey, CA. Evidence Based Research, Vienna, VA.

[14] Russel, S. J. & P. Norvig, 1995, "Artificial Intelligence: A Modern Approach", Prentice Hall.

[15] Kirkpatrick, S. 1984, Journal of Statistical Physics, vol. 34, pp. 975-986.

[16] Vecchi, M.P. & S. Kirkpatrick, 1983, IEEE Transactions on Computer Aided Design, vol. CAD-2, pp. 215-222.

[17] Carley Kathleen M, et al. 2003, "Destabilizing Dynamic Covert Networks" In Proceedings of the 8th International Command and Control Research and Technology Symposium. Conference held at the National Defense War College, Washington DC.

[18] Carley Kathleen M. & Jeff Reminga, 2004, "ORA: Organization Risk Analyzer", Carnegie Mellon University, School of Computer Science, Institute for Software Research International, Technical Report CMU-ISRI-04-101.