# OPTIMIZATION ON THE CRYSTAL MULTICOMPUTER

by

B. Feijoo
and
R. R. Meyer

Computer Sciences Technical Report #562

October 1984

# OPTIMIZATION ON THE CRYSTAL MULTICOMPUTER*

B. Feijoo and R. R. Meyer

Computer Sciences Department and Mathematics Research Center

The University of Wisconsin-Madison

Madison, Wisconsin 53706 USA

## Abstract

Most large-scale optimization problems exhibit special structures that offer the possibility of attack via parallel algorithms. The CRYSTAL multicomputer [DeWitt, et al, 84] under development in the Computer Sciences Department at the University of Wisconsin-Madison offers a unique opportunity to explore distributed approaches to optimization in that it provides a collection of loosely-coupled minicomputers of substantial computing power (specifically, VAX 11/750's) that may exchange data in an efficient manner via messages. In fact, very promising results have already been obtained through the use of CRYSTAL on nonlinear traffic assignment problems [Feijoo and Meyer 83]. For example, a twelve-commodity traffic assignment problem was solved by utilizing twelve processors, one dedicated to each commodity. The solution time using 12 processors was 4.7 times faster than the solution time on a single machine, and it is expected that significant further improvements in the level of parallelism can be achieved in this problem class. The particular piecewise-linear approximation approach to be described below allows the decomposition of these problems at each iteration into a set of individual linear single-commodity network optimization subproblems, which may then be solved in parallel on the CRYSTAL node machines. This method thus has the high level of granularity that is appropriate for CRYSTAL, and appears to be one of the first uses of distributed computing in the solution of constrained optimization problems.

* This research was supported in part by NSF grant MCS8200632 and ARO contract DAAG29-80-C-0041 .

## The CRYSTAL Multicomputer

CRYSTAL is a set of VAX-11/750 computers (currently there are 20) with 2 megabytes of memory each, connected by a 10 megabit/sec Proteon ProNet token ring. It can be used simultaneously by multiple research projects through partitioning the available processors according to the requirements of each project. This partitioning is done via the software, and, once a user has acquired a "partition" or subset of processors, the user then has exclusive access to the node machines of that partition. CRYSTAL software is written in a local extension to Modula. Researchers can employ the CRYSTAL multicomputer in a number of ways. Projects that need direct control of processor resources can be implemented using a reliable communication service (the "nugget") [Cook, et al, 83] that resides on each node processor. Projects that prefer a higher-level interface can be implemented using the Charlotte distributed operating system. The Charlotte kernel provides multiprocessing, inter-process communication, and mechanisms for scheduling, store allocation, and migration. Development, debugging, and execution of projects takes place remotely through any of several VAX-11/780 hosts running Berkeley Unix 4.2. Acquiring a partition of node machines, resetting each node of the partition, and then loading an application onto each node may be performed interactively from any host machine. CRYSTAL has been used for research in a variety of areas, including distributed operating systems, programming languages for distributed systems, tools for debugging distributed systems, multiprocessor database machines, protocols for high performance local network communications, and numerical methods. Future plans for CRYSTAL include an increase in the number of node machines to approximately 40 and an upgrade of the communications medium to an 80 megabit/sec token ring. The Crystal project is supported by a National Science Foundation Coordinated Experimental Research (CER) grant.

## Nonlinear Networks

Nonlinear optimization problems involving network constraints arise in a great variety of applications and are notable for their very large size. Applications areas include computer network design [Cantor and Gerla 74], [Gavish and Hantler 82], [Magnanti and Wong 84], urban traffic assignment [Bertsekas and Gafni 82], [Dantzig, et al, 79], [Lawphongpanich and Hearn 83], [Pang and Yu 82], hydroelectric power systems [Hanscom, et al, 80], [Rosenthal 81], telecommunications networks [McCallum 76], and water supply systems [Beck, et al, 83]. These problems are among the largest nonlinear programming problems that are currently being studied because large-scale networks of various sorts not only arise naturally in many different areas and represent a type of model that people in a variety of disciplines find intuitively easy to develop and maintain, but also exhibit a mathematical structure that can be exploited to develop algorithms that in many cases can solve at affordable cost problems involving tens of thousands of variables (In the case of linear networks, [Barr and Turner 81] describe the solution of problems with 50,000 constraints and 65 million variables being run on a "production basis" for the U. S. Department of the Treasury).

Of particular interest in terms of widespread applicability are problems of traffic routing, equilibrium, and network design in computer and urban transportation networks. These problems are typically multi-commodity problems, i.e., they involve many different types of "commodities" flowing through the network, where depending on the application, a "commodity" will be associated with the traffic flowing out of a "source" node or between a designated origin-destination pair. Enormous problem sizes result from the fact that the number of variables and constraints is determined by the number of links and nodes multiplied by the number of commodities. These same features, however, also make these problems ideal candidates for solution via parallel algorithms, about which more will be said below.

The problems we will consider are of the form:

$$\begin{aligned} &\min f(\mathbf{x}) \\ &s.t.\ \mathbf{A}\mathbf{x} = \mathbf{b} \\ &\qquad \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \end{aligned} \tag{NLP}$$

where $\mathbf{x}$ is an n-vector of flows, $f$ is convex on the n-dimensional interval $[\mathbf{l}, \mathbf{u}]$, and $\mathbf{A}\mathbf{x} = \mathbf{b}$ represents a system of network constraints.

There are two fundamental types of traffic assignment problems: symmetric problems in which the congestion on a given link is determined by the total flow summed over all commodities in that link only, giving rise to a symmetric Jacobian matrix in the corresponding variational problem, and asymmetric problems in which the congestion on a link depends on the total flows in several links. It is well known [Steenbrink 74] that the former problem is equivalent to a convex optimization problem under relatively weak assumptions, whereas the latter gives rise to a variational inequality that has a more complex optimization formulation involving a non-differentiable objective function. In this paper we will concentrate on the former problem, although some of the decomposition ideas considered apply equally well to both problem classes. This problem is of the form (NLP), where the objective function represents total congestion over all of the links of the network, and the constraints represent the supply, demand, and conservation constraints for each of the individual commodities flowing through the network. The number of commodities may be quite large because, depending on the formulation, there can be a commodity corresponding to each node or to each origin-destination pair. However, in the urban traffic equilibrium problem it is typically the case that the only coupling between different commodities occurs in the objective function. Because of this property, the approximation of the objective function by a separable function leads to a decomposition of the problem into separate optimization problems, one for each commodity. These single commodity problems may then be optimized in parallel (many algorithms for the asymmetric problem also contain a phase in which the objective function is replaced by a linear approximation so that the same decomposition may be used). Details are given in the following section.

Turning to a slightly different application, traffic assignment problems in computer networks involving non-bifurcation constraints [Gavish and Hantler 82], as well as network design problems involving the selection of route capacities from a set of discrete alternatives [Magnanti and Wong 84] are non-convex optimization problems requiring the tools of global optimization, and thus are natural candidates for solution on a distributed system. Issues related to global optimization are also discussed below.

## Parallel Algorithms

While the traffic assignment problem has been the subject of investigation by numerous researchers in recent years, and many algorithms have been proposed and tested for its solution, in almost all cases these algorithms lead to a decomposition of the problem into smaller subproblems, and thus are suitable candidates for research on the use of distributed systems. At one end of the spectrum of techniques for this problem class lies the Frank-Wolfe method. in which the original objective function is replaced by a simple linear approximation. The Frank-Wolfe method has the disadvantage that it converges very slowly, but the advantage that a very large degree of parallelism may be achieved.

More complex algorithms include those based on iterative piecewise-linear approximation of the objective function [Feijoo and Meyer 83] and simplicial decomposition ([Lawphongpanich and Hearn 83] and [Pang and Yu 82] in which the feasible set is replaced by a convex combination of feasible points. Computational results have been obtained on the CRYSTAL multicomputer with the piecewise-linear approximation method, so we will consider the details of this approach.

For simplicity we will assume that a feasible solution $\mathbf{x}^i$ (one satisfying all of the constraints) is available at the start of the $i$th iteration of the problem (NLP). (At the initial iteration an arbitrary starting point within $[\mathbf{l}, \mathbf{u}]$ plays the same role.) To simplify notation below,consider the *shifted function* $f_{\mathbf{x}^i}(\mathbf{x}) := f(\mathbf{x}) - f(\mathbf{x}^i)$; it is clear that minimizing $f_{\mathbf{x}^i}$ is equivalent to minimizing $f$. A piecewise-linear separable approximation $\tilde{f}$ "centered" at $\mathbf{x}^i$ is then constructed as follows:

Consider the function

$$h : \mathbf{R}^n \longrightarrow \mathbf{R}$$
$$h(\mathbf{y}) = f_{\mathbf{x}^i}(\mathbf{y} + \mathbf{x}^i)$$

which is a translation of $f_{\mathbf{x}^i}$ to the origin, and let $h_j$ be the restriction of $h$ to the axis $y_j$, that is $h_j(y_j) = h(y_j \mathbf{e}^j)$ where $\mathbf{e}^j$ is the $j$th canonical unit vector. For a given *grid size* $\lambda_j^i$, generate a piecewise-linear approximation $\tilde{h}_j$ to $h_j$, with breakpoints at $\ldots, -2\lambda_j^i, -\lambda_j^i, 0, \lambda_j^i, 2\lambda_j^i, \ldots$ Defining $\tilde{f}_j(x_j) := \tilde{h}_j(x_j - x_j^i)$, the resulting piecewise-linear approximation to $f_{\mathbf{x}^i}$ is:

$$\tilde{f}(\mathbf{x}) = \sum_{j=1}^{n} \tilde{f}_j(x_j)$$

Note that the computation of the $\tilde{f}_j$'s may be carried out *in parallel*. The original objective function is then replaced by the approximating function, and the resulting approximating problem is solved. This problem has the form:

$$\begin{aligned} &\min\ \tilde{f}(\mathbf{x}) \\ &s.t.\ \mathbf{A}\mathbf{x} = \mathbf{b} \qquad\qquad (\mathrm{PLP}(i)) \\ &\mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \end{aligned}$$

Note that the function $\tilde{f}$ depends on the base point $\mathbf{x}^i$ and consequently varies from iteration to iteration.

The problem (PLP($i$)) has two key features: 1) because the objective function is now separable, the problem may be decomposed into $K$ separate optimization problems, where $K$ is the number of commodities in the original problem; and 2) because of the convexity of the original objective function, each of these new problems is equivalent to a linear network optimization problem.

Specifically, assume that $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_K)$, where $\mathbf{x}_k$ is the vector of arcs corresponding to commodity $k$. Since there is no coupling between commodities in the constraints, the problem (PLP($i$)) may be solved by solving *in parallel* (via very fast linear network codes) the set of problems:

$$\begin{aligned} &\min \sum_{j \in S_k} \tilde{f}_j(x_j) \\ &s.t.\ \mathbf{A}_k \mathbf{x}_k = \mathbf{b}_k \qquad\qquad k = 1, \ldots, K \\ &\mathbf{l}_k \leq \mathbf{x}_k \leq \mathbf{u}_k \end{aligned}$$

where $\mathbf{A}_k$, $\mathbf{b}_k$, $\mathbf{l}_k$ and $\mathbf{u}_k$ are for commodity $k$ the corresponding components of $\mathbf{A}, \mathbf{b}, \mathbf{l}$ and $\mathbf{u}$.

Given the set of solutions of these problems $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_K$, if we let $\tilde{\mathbf{x}}^{i+1} = (\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_K)$, the next iterate $\mathbf{x}^{i+1}$ is obtained by solving the line search problem:

$$\begin{aligned} &\min_{\theta}\ f(\mathbf{x}^i + \theta(\tilde{\mathbf{x}}^{i+1} - \mathbf{x}^i)) \\ &s.t.\ \mathbf{l} \leq \mathbf{x}^i + \theta(\tilde{\mathbf{x}}^{i+1} - \mathbf{x}^i) \leq \mathbf{u} \\ &\theta \geq 0 \end{aligned}$$

Once the next iterate has been obtained, the procedure is repeated with a reduced grid size of $\lambda^{j+1} = \alpha\lambda^j$ , where $\alpha \in (0,1)$ ($\alpha = 0.25$ has worked well on our tests).

It has been shown that this procedure will yield a sequence of iterates whose objective function values converge to the optimal value of the original problem (NLP).

## Computational Results

Our current series of tests have been limited to small problems in which the number of commodities is not greater than the number of available processors. Procedures for dealing with larger numbers of commodities will be discussed in the next section. Thus far, three test problems have been used. All three are of the form of urban traffic equilibrium problems. The sources of these problems are: Problem 1, [Pang 83]; Problem 2, [Bertsekas and Gafni 82]; and Problem 3, [Steenbrink 74].

Let us now consider the implementation of the parallel algorithm described above on CRYSTAL. First a partition of node machines is obtained. The number of machines corresponds to the number of commodities of the particular problem. Node machine 1 is used as what we call a master node, machines 2 through $K$ are called slave nodes. A master program is linked and loaded onto node 1 and a slave program onto nodes 2 through $K$. The host program is run in the host machine (a VAX-11/780), which reads all the data and sends it to the master node. The master node then distributes the appropriate subproblem data to the slave nodes and each node (including the master) solves its corresponding subproblem via the RNET linear network optimization code [Grigoriadis 82]. Once the master node finishes solving its subproblem, it waits, if necessary, for the slave nodes to end their computations, then receives the solution data from each of them and finally performs the line search to produce an improved feasible solution. This process is repeated for as many iterations as required to satisfy the stopping criteria. Finally the master node sends back to the host machine the solution and the timing data.

Table 1 shows computing times for the test problems. They were solved also in a sequential fashion on a VAX-11/750 for purposes of comparison. A description of the nature of the objective function $f$ is included for each problem, where $f_l$ represents *total* flow on link $l$ (i.e. the sum of the flows of commodities that use that link). For the CPU times in the parallel case, two times appear, the first is the CPU time for the master node and the second is an average of the times of the slave nodes. Note that the master node performs the line search routine and thus has a higher CPU time than the slave nodes. It should be remarked that in these particular examples the CPU time for each slave node does not differ much from the average. The difference between clock and CPU time for the master node is due to waiting and communication times.

**Problem 1**

*56 variables, 40 constraints, 2 commodities*

$$f = \sum_l a_l f_l^5 + b_l f_l$$

| | Parallel | Sequential |
|---|---|---|
| CPU time | 13.0 s./3.6 s. | 16.8 s. |
| Clock time | 15.2s. | 17.2 s. |

**Problem 2**

*60 variables, 60 constraints, 5 commodities*

$$f = \sum_l \frac{f_l^3}{3} + \frac{f_l^2}{2} + f_l$$

| | Parallel | Sequential |
|---|---|---|
| CPU time | 4.8 s./1.0 s. | 8.5 s. |
| Clock time | 7.0 s. | 8.9 s. |

**Problem 3**

*432 variables, 108 constraints, 12 commodities*

$$f = \sum_l a_l f_l^2 + b_l f_l$$

| | Parallel | Sequential |
|---|---|---|
| CPU time | 13.7./8.5 s. | 111.0 s. |
| Clock time | 23.5 s. | 111.8 s. |

**TABLE 1: Computing times**

## Further Research

For large-scale traffic assignment problems an important area of research in parallel computation will be the distribution of the subproblems among the available processors. In the case of CRYSTAL, in which the number of processors will be approximately 40, there will be a shortage of processors since such problems may contain hundreds or thousands of commodities, each of which could be optimized at a particular iteration on a single processor. It will thus be appropriate to investigate the relative efficiencies of solving all of the single commodity problems at a given iteration by distributing them in fixed groups among the available processors versus having a dynamic allocation mechanism that assigns single commodity problems to newly idle processors versus solving only a proper subset of the single commodity problems and then doing the line search restricted to the corresponding subset of variables while the processors are working on another subset of the single commodity subproblems. The latter procedure has the advantage of increasing parallelism , but leads to interesting mathematical questions in that it may be necessary to change the objective function of the optimization problems during or after the solution process as a result of the line search. In a sense, traffic assignment problems are ideal for experiments in parallel computing since they involve large-scale problems and allow the testing of a wide variety of strategies for mapping processes to processors.

Problems of global optimization also lend themselves readily to the development of parallel algorithms, since the fundamental difficulty associated with global optimization is the necessity of ensuring that the behavior of the objective function in all subregions of the feasible set is accounted for, and to accomplish this it is usually necessary to deal separately with a number of subregions. Two recent methods of global optimization that are promising and particularly well suited for research on a distributed system are those of [Rosen 83] and [Kan and Timmer 84] (see also [Schnabel 84] for a recent survey of parallel computing in optimization). In Rosen's method, which is deterministic, there is an initial stage in which a multiple-cost-row linear program with 2n cost vectors is solved, followed by an LP solution, and then a final branch-and-bound stage. It would be of interest to investigate strategies for solving in parallel the multiple-cost-row row LP, trying to distinguish between those objective functions that are sufficiently different so as to merit separate optimizations from those most efficiently dealt with by means of post-optimal analysis of a "related" objective function. In the branch-and-bound phase there are interesting issues of how to decompose problems so as to make maximum use of the available processors. A traditional binary branching will result in an initial delay before all processors are being utilized, particularly in the case of large subproblems. On the other hand, a finer decomposition may result in the consideration of uninteresting subproblems. A study of mechanisms to identify potentially interesting subproblems, through the use of estimation procedures for optimal objective values over subregions, is clearly in order. These observations also apply to branch-and-bound procedures arising from other problem contexts.

Kan has proposed a stochastic method for unconstrained global optimization that initially involves the computation of function values at a number of trial points generated via a uniform distribution over an n-dimensional box known to contain a global optimum in its interior, followed by a clustering operation whose goal is to identify regions of attraction of local minima, followed by local searches from starting points identified by the clustering. This technique has nice statistical properties and has performed well on test problems with small numbers of variables. All of its procedures are suited to parallel computation, and the clustering and starting point selection phases are areas in which much research remains to be done. It will also be worthwhile to investigate the simultaneous use of different local search algorithms from selected starting points,and to consider methods of parallelizable random search, such as those proposed by [Aluffi-Pentini, et al 83].

## Conclusions

Recent developments in distributed computing systems have made possible realistic experiments with parallel algorithms in optimization. The CRYSTAL multicomputer in particular has proved to be an effective tool for research in parallel methods for multicommodity traffic assignment problems, and is also promising for studies in global optimization.

## References

F. Aluffi-Pentini,V.Parisi,and F.Zirilli:"Global optimization and stochastic differential equations",Technical Report,Istituto Matematico, Università di Camerino, Camerino, Italy, 1983.

R. S. Barr and J. S. Turner: "Microdata file merging through large-scale network technology", *Mathematical Programming Study 15* , 1981, 1-22.

D. P. Bertsekas and E.M.Gafni:"Projection methods for minimum cost network flow problems", *Mathematical Programming Study 17*, 1982,139-159.

D. G. Cantor and M. Gerla: "Optimal routing in packet switched computer networks", *IEEE Transactions on Computing* C-23, 1974, 1062-1068.

R. Cook, R. Finkel, D. DeWitt, L. Landweber, T. Virgilio: "The CRYSTAL nugget", Technical Report 499, Computer Sciences Department, The University of Wisconsin-Madison, April, 1983.

G. Dantzig, R. Harvey, Z. Lansdowne, D. Robinson, and S. Maier: "Formulating and solving the network design problem by decomposition", *Transportation Research* 13B, 1979, 5-17.

D. DeWitt, R. Finkel, and M. Solomon: "The CRYSTAL multicomputer: design and implementation experience", Technical Report 553, Computer Sciences Department, The University of Wisconsin-Madison, September, 1984.

B. Feijõo and R. R. Meyer: "Piecewise-linear approximation methods for nonseparable convex optimization", Technical Report 521, Computer Sciences Department, The University of Wisconsin-Madison, 1983.

B. Gavish and S. L. Hantler: "An algorithm for optimal route selection in SNA networks", Research Report RC 9549, IBM T. J. Watson Research Center, Yorktown Heights, N. Y., August, 1982.

M. D. Grigoriadis: "Minimum-cost network flows, part 1: an implementation of the network simplex method", Technical Report LCSR-TR-37, Laboratory for Computer Science Research, Rutgers University, New Brunswick, N. J., September, 1982.

M. A. Hanscom, L. Lafond, L. Lasdon and G. Pronovost: "Modeling and resolution of the medium term energy planning problem for a large hydro-electric system", *Management Science* 26, 1980, 659-668.

A. H. G. Rinooy Kan and G. T. Timmer: "Stochastic Methods for Global optimization", paper presented at the NATO Advanced Study Institute on Computational Mathematical Programming, Bad Windsheim, W. Germany, July, 1984.

S. Lawphongpanich and D. W. Hearn: "Restricted Simplicial Decomposition with Application to the traffic assignment problem", Research Report 83-8, Department of Industrial and Systems Engineering, University of Florida, Gainesville, Fl., September, 1983.

T. L. Magnanti and R. T. Wong: "Network design and transportation planning: models and algorithms", *Transportation Science* 18, 1984, 1-55.

C. J. McCallum: "A Generalized Upper Bounding approach to Communications network planning problems", *Networks* 7, 1976, 1-23.

J. S. Pang, private communication, 1983.

J. S. Pang and C. S. Yu: "Linearized simplicial decomposition methods for computing traffic equilibria on networks", Technical Report, University of Texas at Dallas, Richardson, Texas, 1982.

J. B. Rosen: "Computational experience with large-scale constrained global optimization", *Committee on Algorithms Newsletter*, March, 1984.

R. E. Rosenthal: "A nonlinear network flow algorithm for maximization of benefits in a hydroelectric power system", *Operations Research* 29, 1981, 763-786.

R. B. Schnabel: "Parallel computing in optimization", paper presented at the Nato Advanced Study Institute on Computational Mathematical Programming, Bad Windsheim, West Germany, July, 1984.

P. A. Steenbrink: *Optimization of Transport Networks*, Wiley, London, 1974.