Simulation and implementation of the Parallel Control Random Search Algorithm[1]

J. Balogh, P.M. Ortigosa, and I. García

In this work we propose to study the Controlled Random Search (CRS) algorithm of Price [1, 2] in order to get an efficient parallel version. Some parallel approaches have been proposed by García [3, 4], McKeown [5], Sutti [6], Ducksbury [7], Price [8] and Woodhams and Price [9] using several kind of parallel computers and strategies.

The Price sequential procedure to globally minimize $f : R^n \to R$ can be summarized as follows:

1. Generate N random points $R_0, \ldots, R_{N-1}$ in the feasible region (F.R.) and evaluate f in each point.
2. Find the best point $(R_B, f(R_B))$ and the worst one $(R_W, f(R_W))$ among $x_1, \ldots, x_N$, such that $f(R_B) \leq f(R_k)$, and $f(R_W) \geq f(R_k)$, for all k.
3. Choose randomly $n+1$ points $R^0, \ldots, R^n$. Compute $\overline{P} = 2 \times \overline{G} - \overline{R}^n$, where $\overline{G}$ is the centroid for $R_0, \ldots, R_{n-1}$. If $\overline{P}$ is in the F.R and $f(\overline{P}) \leq f(R_W)$, substitute R_W, with $\overline{P}$in the sample set, otherwise if success rate $> 50\%$, compute $\overline{P} = \frac{\overline{G}+\overline{R}_n}{2}$. If $f(\overline{P}) \leq f(R_W)$ holds for the point $\overline{P}$, then substitute R_W, otherwise repeat step (iii),
4. Stop if the termination criterion is satisfied, otherwise go to step (ii).

In the above scheme, let us distinguish three stages: The first of Initialization (I.S.) consisting of step (i), which occurs once only in the procedure, the second of Update (U.S.) consisting of step (ii) and the third of Exploration (E.S.) consisting of steps (ii) and (iv).

As it can be seen, this algorithm is highly sequential, because each new point $\overline{P}$ is generated from the sample set which consist of the best previous points computed before it. However, it is possible to detect some parallelism. At the initialization stage $R_0, \ldots, R_{N-1}$ can be simultaneously generated, and $f(R_0), \ldots, f(R_{N-1})$ can be simultaneously computed.

So it seems that the best strategy to parallelize this algorithm is the global one, where a master-worker communication model is required. In this strategy the master processor executes the PCRS algorithm and a worker processor works only with the trial points supplied by the master processor. In our strategy the master processor generates the points and sends them to the worker processors. In this way, a worker processor only evaluates the objective function at the trial points supplied by the master processor. After every evaluation it sends the result back to the master [4].

García et al. [10, 4] implemented this strategy. Their algorithm was fully asynchronous except at the end of the initialization stage, where the master did not start to generate $\overline{P}$ points until the initial sample set was evaluated. Once the set was initialized, the master processor generated NP (Number of Processors) trials points and sent them to the workers, so each worker processor received a trial point to evaluate. After every evaluation, workers send the result back to the master and receive a new point to evaluate. Although this approach is asynchronous, it is possible that a worker processor has to wait for a new point if another worker(s) has (have) sent its (their) function evaluation to the master at the same time. It could be solved if the worker processors have more than one trial point to evaluate. So, when a worker processor finishes an evaluation and sends the result to the master, it can go on evaluating another of its points while the master receives the information and sends a new point back to the worker processor. In this way not only the above problem is solved, but also the overhead from the communication cost is reduced, because it is overlapped with some computation cost.

However, another problem arises of this kind of asynchronous parallel versions. Data, which are used in simultaneous processes may belong to different iterative cycles, so the master processor generates new

[1] This work was supported by the Tempus JEP 11264-96, by the Consejeria de Educación de la Junta de Andalucía (07/FSC/MDM) and by the Ministry of Education of Spain (CICYTTIC96-1125-C03-03).

trial points from the sample set which has not all the information about the most recently generated point. This may produce that the new points will not be as good as in the sequential case, and then more trial points have to be generated, and the number of evaluations in the parallel version can be higher than for the sequential one. This problem raised in the García's implementation where the worker processors only had a point to evaluate each time. This increase in the number of function evaluations can even be higher in the case when the worker processors have two or more points to evaluate.

Therefore, it is rather important to know the effects of increasing the number of points that each worker processor stores to send sequencialy to the master processor once one of this stored points is evaluated.

We are changing the sequential version of CRS in order to simulate the behaviour of the new parallel version where a worker processor has some points to evaluate instead of only one. We will study the effects of these changes and determine the amount of trial points that a worker processor must have to get the best performance.

We will implement the parallel version (previously simulated and studied) in a MIMD system using PVM and MPI, and its performance will be analyzed.

A set of test functions will be used to check the convergence and the parallel performance. As this algorithm has a strong stochastic component, the number of function evaluations needed to reach the global minimum depends on the particular execution. For this reason the algorithm will be executed 100 times for each case obtaining a stable statistical sample. From this data set, mean value (μ) of the number of function evaluations and the corresponding confidence intervals (95%) will be computed (see [11]). So, the probability of an interval covering the mean is 0.95 or, expressed in another way, that on the average 95 out of 100 confidence intervals similarly obtained would cover the mean.

References

[1] W.L. Price, "A controlled random search procedure for global optimization," in *Towards Global Optimization 2* (L.C.W Dixon and G.P. Szegö, ed.), pp. 71–84, Amsterdam: North Holland, 1978.

[2] W.L. Price, "Global optimization algorithms by controlled random search," *Journal of Optimisation Theory and Applications*, no. 40, pp. 333–348, 1983.

[3] I. García, G.T. Herman, "Global optimization by parallel constrained biased random search," in *State of Art in Global Optimization: Computational Methods and Applications* (C.A. Floudas and P.M. Pardalos, eds.), Kluwer, Dordrecht, pp. 433-455.

[4] I. García, P.M. Ortigosa, L.G. Casado, G.T.Herman and S. Matej, "Multidimensional Optimization in Image Reconstruction from Projections," *Developments in Global Optimization*, Kluwer, Dordrecht, pp. 289–300, 1997.

[5] J.J. McKeown, "Aspects of parallel computations in numerical optimization," in *Numerical Techniques for Stochastic Systems* (F. Archetti and M. Cugiani, eds.), pp. 297–327, 1980.

[6] C. Sutti, "Local and global optimization by parallel algorithms for MIMD systems," *Annals of Operations Research*, no. 1, pp. 151–164, 1984.

[7] P.G. Duckbury, *Parallel Array Processing.* Chichester: Ellis Horwood, 1986.

[8] W.L. Price, "Global optimization algorithms for a CAD workstation," *Journal of Optimisation Theory and Applications*, no. 55, pp. 133–146, 1987.

[9] F.W.D. Woodhams, W.L. Price, "Optimizing accelerator for CAD workstation," *IEE Proceedings Part E*, vol. 135, no. 4, pp. 214–221, 1988.

[10] I. García, P.M. Ortigosa, L.G. Casado, G.T.Herman and S. Matej, “A parallel implementation of the controlled random search algorithm to optimize an algorithm for reconstruction from projections,” in *IIIrd Workshop on Global Optimization*, (Szeged, Hungary), pp. 28–32, December 1995.

[11] R.R. Sokal and F.J. Rohlf, *Biometry*. New York: Freeman, 1981.