DISCRETE AND GLOBAL OPTIMIZATION IN EVEREST DISTRIBUTED ENVIRONMENT BY LOOSELY COUPLED BRANCH-AND-BOUND SOLVERS

S.A. Smirnov [a], V.V. Voloshinov [b]

Institute for Information Transmission Problems of the Russian Academy of Sciences (Kharkevich Institute), Bolshoy Karetny per. 19, build.1, Moscow, 127051, Russia

E-mail: [a] sasmir@gmail.com, [b] vv_voloshinov@iitp.ru

The report presents an approach to solve hard problems of discrete and global optimization in distributed computing environment Everest – a web-based computing platform, http://everest.distcomp.org. The approach integrates: solvers implementing branch-and-bound (B&B) algorithm; preliminary decomposition of feasible domain and/or B&B multi-search (of the same problem with different settings of B&B algorithm); pool of solvers, which run in parallel and exchange with values of incumbents found by B&B. The report partially sums up results of research and programmatic implementation of coarse-grained (loosely coupled) parallelisation in the form of Everest application called DDBNB, https://github.com/distcomp/ddbnb. On solving hard geometric combinatorial problems (Tammes problem, Thomson problem and Flat Torus Packing problem) fine-grained parallel solver ParaSCIP (based on MPI) has been used also. In particular, computing proof of Flat Torus optimal packing for 9 circles has been obtained. Comparison of ParaSCIP and DDBNB performance leads to future plan to improve DDBNB by integration of coarse- and fine-grained parallelism.

Keywords: distributed computing, Everest platform, discrete and global optimization, feasible domain decomposition, coarse-grained parallelism, Tammes problem, Thomson problem, Flat Torus Packing.

1. Coarse-grained parallelization and loosely-coupled distributed application

The report presents an new approach to solving rather hard discrete and global optimization problems in Everest, http://everest.distcomp.org, a web-based distributed computing platform [1]. The platform enables convenient access to heterogeneous resources (standalone servers, high performance clusters etc.) by means of domain-specific computational web services, development and execution of many-task applications, and pooling of multiple resources for running distributed computations. Rather generic Everst-application had been implemented for solving discrete and global optimization problems - so called DDBNB, Domain Decomposition Branch-and-Bound, https://github.com/distcomp/ddbnb. DDBNB implements a kind of coarse-grained parallelization of Branch-and-Bound (B&B) algorithm. It supports two strategies (including combined usage of both): decomposition of feasible domain into a set of sub-problems; multisearch (or concurrent) solving the same problem with different settings of the B&B-method. In both cases several B&B-solver's processes exchange incumbents, best values of goal function on feasible solutions, they found. DDBNB uses generic Everest messaging service and open source solvers: CBC, https://projects.coin-or.org/Cbc; SCIP, http://scip.zib.de. So DDBNB is an example of loosely-coupled parallel application based on non-intensive data exchange [2, 3]. One should note that DDBNB programmatic implementation inherits generic Everest Parameter Sweep application [4].

For all optimization problems considered below formulation of original problems and domain decomposition strategies have been implemented in Python via Pyomo (Python Optimization Modeling Objects), http://www.pyomo.org. All sub-problems have been passed to solvers as AMPL-stubs, (also known as NL-files). Usage of these high level optimization modeling tools gives opportunity to try a number of empirical rules for domain decomposition strategies.

2. Hybridization of domain decomposition and concurrent mode of parallelization

Paper [2] presents a coarse-grain parallelization of B&B-algorithm that is based on a preliminary decomposition of a feasible domain of a MILP (Mixed-Integer Linear Programming) problem by some heuristic rules. Subproblems produced by the domain decomposition are distributed across a pool of standalone BnB-solvers running in parallel. The incumbent values found individually by each solver are intercepted and propagated to other solvers in order to speed up the traversal of BnB search tree. This mode of DDBNB operation may be called as *domain decomposition*. Another approach to coarse-grain parallelism has been described in article [3]. It is similar to what was implemented previously in DDBNB, but without domain decomposition. Instead, different BnB-processes work in parallel with the same optimization problem, but with different settings of B&B-algorithm. This approach is known in the literature as *concurrent parallelization*, *portfolio parallelization* or *multi-search* (see references in [3]).

Current implementation of DDBNB support hybridization of two modes mentioned above: domain decomposition gives a number of subproblems and each of subproblems are solving with different B&B-settings. So if we have N_{dd} subproblems and N_s sets of B&B options DDBNB will run $N_{dd}* N_s$ B&B-solvers. Below we describe an example of hybrid mode usage for Traveling Salesman Problem (TSP) formulated as MINLP (see details in [2, 3]).

There were prepared a number of randomly generated TSP for $N=70, 80, 90$, where N – is the number of graph vertices ("cities" to be visited by Salesman). For every N six random problems hade been generated. Domain decomposition has been done by the following rule: all arcs $\{i,j\}$ (a path between *i-th* and *j-th* vertices) have been sorted in the ascending order of the length $d(i,j)$ of the path; then first K of boolean variables $x(i_k,j_k)$ have been fixed to 1 or 0 (including or excluding the arc $\{i_k,j_k\}$ from optimal path). So, for given $K=0,1,2,3,4$, $N_{dd}=2^K$ subproblems have been generated. For short let's refer them as DD-subproblems.

As to concurrent mode, SCIP solver has more than 2000 parameters might be set by users. Only one of them, *nodeselection/childsel*, has been chosen. It defines the rule of child node selection while searching B&B-tree. One of the following values can be used (see description in [3]): *'d'own*, *'u'p*, *'p'seudo costs*, *'i'nference*, *'l'p value*, *'r'oot LP value*, *'h'ybrid*. Thus every subproblems can be solved in parallel by seven solvers with different values of nodeselection/childsel, $N_s=7$.

So, for every *N* there were performed 6 experiments (see Figure 1): “SCIP” as a reference running without any parallelisation; “0” - only Concurrent mode, 7 solvers run in parallel; “1” - DD by one boolean variables, two subproblems * 7 settings -14 solvers run in parallel; “2” - 4*7=28 solver processes; “3” - 8*7=56 solver processes; “4” - 16*7=112 solver processes. Vertical columns in Figure 1 are sums of solution times for all 6 experiments for a given *N*. You can see that, in average, the best performance have been obtained when a pair of DD-subproblems were solved in parallel by 7 SCIP solvers with different B&B-settings.

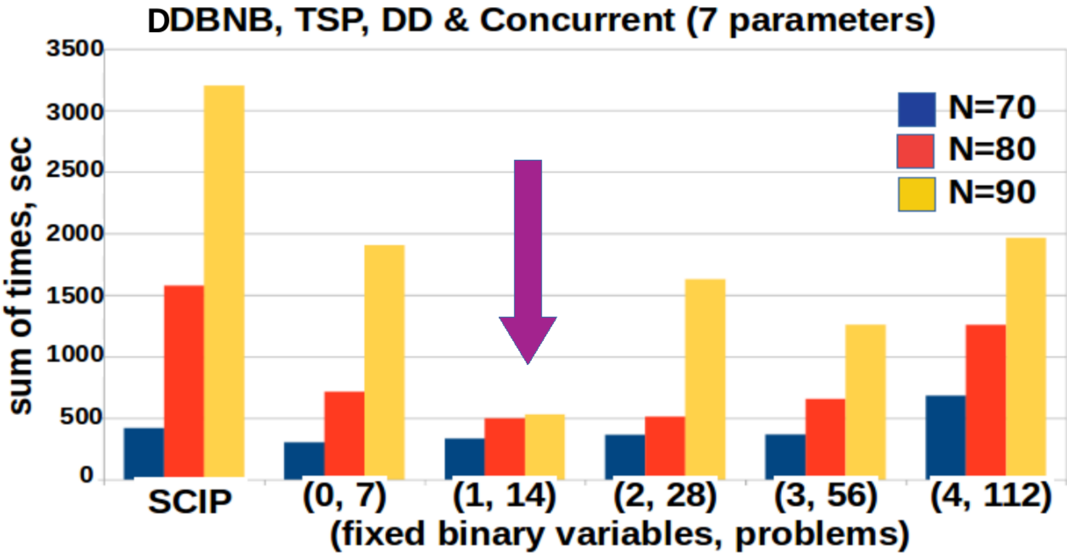

Figure 1. Experiment with hybrid mode of DDBNB for Traveling Salesman Problems

3. Geometric combinatorial problems as global optimization

A number of experiments have been done with various geometric combinatorial problems: Tammes problem, Thomson problem and Flat Torus Packing problem (FTPP) [5, 6, 7, 8]. All of them has been converted to mathematical programming problems with polynomials in constraints.

Tammes problem (to maximize the minimal distance between any pair of N points on 3D-sphere) is formulated as following

$$\min_{1\le i<j\le N}\left\|x_i - x_j\right\| \rightarrow \min_{x_i\in R^3, i=1:N} : s.t.\ \left\|x_i\right\| = 1,\ i = 1:N \qquad \textbf{(1)}$$

It is easily seen that Tammes problem is equivalent to the following ()

$$\begin{gathered} z \rightarrow \min_{z,x_i\in R^3, i=1:N}, \\ x_i^{\mathrm{T}} x_j = \sum_{k=1:3} x_{i,k}x_{j,k} \le z \ \ (1\le i < j \le N), \\ \left\|x_i\right\| = \sum_{k=1:3}\left(x_{i,k}\right)^2 = 1 \ \ (i = 1:N) \end{gathered} \qquad \textbf{(2)}$$

Thomson problem (to minimize electrostatic Coulomb energy of N unit charges put on a sphere) in original form (3) and as equivalent NLP with 4$^{\text{th}}$ order polynomials (4)

$$\min_{1\le i<j\le N}\left\|x_i - x_j\right\|^{-1} \rightarrow \min_{x_i\in R^3, i=1:N} : s.t.\ \left\|x_i\right\| = 1,\ i = 1:N \qquad \textbf{(3)}$$

$$\sum_{1\le i<j\le N} z_{ij} \to \min_{z,\, x_i \in R^3,\, i=1:N},$$
$$z_{ij}^2 \sum_{k=1:3}(x_{i,k} - x_{j,k})^2 = 1 \ \ (1 \le i < j \le N), \tag{4}$$
$$\|x_i\| = \sum_{k=1:3}(x_{i,k})^2 = 1 \ \ (i = 1:N)$$

A number of auxiliary constraints have been added to reduce volume of the feasible domain in both above problems by removing redundant solutions that might be obtained by 3D rotations, mirror transformations and renumbering of points. Typical ones are presented below.

$$x_{1,1} = x_{1,2} = 0, x_{1,3} = 1,$$
$$x_{2,1} = 0,\ x_{2,2} \ge 0,$$
$$x_{i+1,3} \le x_{i,3} \ \ (1 \le i \le N-1)$$

Flat Torus Packing problem (see [8]) in its original form (5) is based in the following special distance on 2D-plane defined in (6).

$$z \to \min_{z,\, x_i \in R^2,\, i=1:N},$$
$$z \le d(x_i, x_j) \ \ (1 \le i < j \le N), \tag{5}$$
$$0 \le x_{i,k} \le 1 \ \ (k = 1:2,\ i = 1:N)$$

$$\mathrm{d}(x, y) = \sqrt{\sum_{k=1,2}\left(\min\left\{|x_k - y_k|, 1 - |x_k - y_k|\right\}\right)^2} \tag{6}$$

Representation of (5) as MINLP is rather cumbersome to write it here, see [8] for details.

All above problems may be solved by SCIP solver [9] integrated to DDBNB. A number of results concerning solving Tammes and Thomson problems by DDBNB are presented in [5, 6]. The new results concern usage of ParaSCIP solver that is a fine-grained (MPI-based) parallel implementation of B&B algorithm [10]. A few experiments have been done in our attempts to use ParaSCIP. It enables to compare performance of DDBNB and ParaSCIP. For example, see [5], Thomson problem for N=5, after decomposition into 16 subproblems, has been solved by DDBNB on HPC4 KI ("Kurchatov institute") cluster in 160 minutes (by 16 SCIP processes running on clusters working nodes), when one SCIP process running on one CPU took 600 minutes. With the same problem ParaSCIP was 4 times more effective than DDBNB: 86 min. on 8 CPUs.

With FTPP ParaSCIP solvers also gave promising results. E.g. FTPP, N=8, has been solved by one SCIP process (on HPC4 KI) in 780 minutes. ParaSCIP used 8 CPUs and solved the same problem in 126 minutes, which corresponds to 0.77 efficiency of parallelization. And the most important result [8]: ParaSCIP gave numerical proof of conjecture about optimal FTPP solution for N=9, which was presented in [7].

5. Acknowledgement

The work was supported by the Russian Foundation for Basic Research (RFBR), grants #18-07-01175 "Development of computing methods based on solution of loosely-coupled optimization problems in heterogeneous computing environment" and #18-07-00956 "The development of methods and tools for distributed computing on the base of Everest platform". The most intensive calculations has been carried out using computing resources of the federal collective usage center Complex for Sim-ulation and Data Processing for Mega-science Facilities at NRC "Kurchatov Institute" (ministry subvention underagreement RFMEFI62117X0016), ckp.nrcki.ru.

6. Conclusion and future plans

Coarse-grained parallelization of B&B-algorithm has been implemented as DDBNB Everest application via integration of simple Everest message service, generic Parameter Sweep Everest application and open source B&B-solvers (SCIP and COIN-OR CBC). Two modes of DDBNB operation (domain decomposition and multi-search) have been tested by randomly generated Traveling Salesman Problems and demonstrated rather good performance. DDBNB may be run in hybrid mode when original problem is decomposed into subproblems solving with different B&B-settings. Proposed coarse-grained and loosely-coupled (in data exchange) approach, though it is very simple, may be useful if domain decomposition has been done properly. One should mention that the choice of decomposition rules remains an open issue as for discrete problems and for global optimization as well. E.g., for global optimization with polynomials in constraints, reduction of volume of feasible domain usually gives significant speed-up.

DDBNB Everest application, https://github.com/distcomp/ddbnb, became rather mature and provides use of domain decomposition and/or concurrent mode of B&B parallelization in heterogeneous computing environment. Main drawback of DDBNB is absence of dynamic workload balance between available resources. To solve this issue we plan to integrate MPI-based ParaSCIP into DDNBN. On success, it permits to use a few different clusters in solving one hard optimization problem. Thus, researchers will get an integration of coarse- and fine-grained parallelism in heterogeneous computing environments.

References

[1] Sukhoroslov O., Volkov S., Afanasiev A. A Web-Based Platform for Publication and Distributed Execution of Computing Applications // 14th International Symposium on Parallel and Distributed Computing (ISPDC). IEEE, 2015, pp. 175-184.

[2] Voloshinov V., Smirnov S. and Sukhoroslov, O., Implementation and Use of Coarse-grained Parallel Branch-and-bound in Everest Distributed Environment // Procedia Computer Science (ISSN: 1877-0509), 108, 2017, pp. 1532-1541. DOI: 10.1016/j.procs.2017.05.207.

[3] Smirnov S., Voloshinov V. Implementation of Concurrent Parallelization of Branch-and-bound algorithm in Everest Distributed Environment // Procedia Computer Science, 119, 2017, pp. 83–89 DOI: 10.1016/j.procs.2017.11.163.

[4] Volkov S., Sukhoroslov O. A Generic Web Service for Running Parameter Sweep Experiments in Distributed Computing Environment. Procedia Computer Science, Volume 66, 2015, pp. 477-486. DOI: 10.1016/j.procs.2015.11.054

[5] Smirnov S., Voloshinov V. On Domain Decomposition Strategies to Parallelize Branch-and-Bound Method for Global Optimization in Everest Distributed Environment // Procedia Computer Science, 136C, 2018, vol. 136, pp. 128-135. DOI: 10.1016/j.procs.2018.08.245.

[6] Smirnov S., Sukhoroslov O. and Voloshinov V. Using Resources of Supercomputing Centers with Everest Platform // Russian Supercomputing Days: Proceedings of the international conference (September 24-25, 2018, Moscow,Russia), Moscow State University, 2018, pp. 649-660

[7] O. R. Musin and A. V. Nikitenko. Optimal packings of congruent circles on a square flat torus. Discrete & Computational Geometry, vol. 55, No. 1, pp. 1–20, 2016.

[8] Smirnov S.A., Voloshinov V.V. Packing of Circles on Square Flat Torus as Global Optimization of Mixed Integer Nonlinear problem // arXiv preprint arXiv:1809.10525 , 2018.

[9] A. Gleixner, M. Bastubbe, L. Eifler, T. Gally, G. Gamrath, R. L. Gottwald, G. Hendel, C. Hojny, T. Koch, M. E. L¨ubbecke, S. J. Maher, M. Miltenberger, B. M¨uller, M. E. Pfetsch, C. Puchert, D. Rehfeldt, F. Schl¨osser, C. Schubert, F. Serrano, Y. Shinano, J. M. Viernickel, M. Walter, F. Wegscheider, J. T. Witt, and J. Witzig. The SCIP Optimization Suite 6.0. Technical Report 18-26, ZIB, Takustr. 7, 14195 Berlin, 2018.

[10] Yuji Shinano, Tobias Achterberg, Timo Berthold, Stefan Heinz, and Thorsten Koch. ParaSCIP: a parallel extension of SCIP. In Competence in High Performance Computing 2010, pp. 135–148. Springer, 2011.