Memory Access Optimization for Computations on Unstructured Meshes

Thesis submitted for the degree of Doctor of Philosophy

Antal Hiba M.Sc.

Supervisors:

Dr. Péter Szolgay, Dr. Miklós Ruszinkó

Pázmány Péter Catholic University

Faculty of Information Technology and Bionics

Roska Tamás Doctoral School of Sciences and Technology

Budapest, 2015

Motivation and scope

Recent processor architectures consist many parallel cores. These chips with high computational capacity are common and also the building blocks of current scientific and industrial supercomputers. The theoretical computational power reached 1 Tera-FLOPS/chip, but the utilization of such processors can be just 10-15% in a real-life application. The common reason is the difference between the memory bandwidth and the computational capacity of the processor. The speed of computation is not limited by the processing elements (PE), rather the memory bandwidth, which becomes a kind of wall between the processor and the memory [R1]. This effect is getting stronger because the computational capacity increases faster than the memory bandwidth. Several projects aim for a solution to the memory wall [R2] from these approaches I prefer the integration of memory with the processor array with 3D chip technology [R3, R4]. Memory interface is not only a speed limiting factor, it also increases the power consumption thus the optimized usage of these interfaces becomes important for economical and green computing reasons too.

Memory access optimization means the local memory (cache) handling methods of the processor, which decreases the number of transfers from/to the slow off-chip memory, and includes also the methods, which support the utilization of available off-chip memory bandwidth. The number of data transfers can be decreased by the on-chip cache, if a data element can be reused during the computation. The utilization of available off-chip memory bandwidth can be increased by optimized memory access patterns. For the current DRAM technology the serial access pattern is the most appropriate thus it is beneficial to organize the input data to reach serial access patterns during the computation. The task becomes more complex when the input problem is distributed among multiple processor chips. In this case, the inter-processor communication comes into the scene, which has to

be considered to reach the best possible performance.

Classical distribution approaches consider only the interprocessor communication and identical part sizes. The elementary operators inside a task define a graph, where the nodes represent elementary operators, and the edges describe data dependencies. At this point, we reach the graph partitioning problem, which aims for identically sized subgraphs with minimal edge cut (communication need). The optimization of memory access is done by a node ordering algorithm, which increases data locality after the partitioning phase. Data locality maximization leads to the Matrix Bandwidth Minimization problem in which a matrix is transformed into a narrow banded form. Unfortunately, Graph Partitioning and Matrix Bandwidth Minimization are NP-complete [R5, R6] thus an optimal solution can not be obtained in polinomial time assuming P!=NP. Thanks to many successful research attempts there are several good heuristics to handle these problems. Because data locality becomes more and more important, a question arises: Should we consider data locality at the partitioning phase?

The primary goal of this dissertation is to investigate the connections between the partitions and the reachable data locality, and based on this knowledge, construct methods which can consider data locality and inter-processor communication at the same time. This approach can ensure better processor utilization and evade the wasting of resources. Parallelization and better memory bandwidth utilization require the solution of complex optimization tasks. Surely we do not want to spend much time with these optimization tasks. Novel efficient heuristics are often called metaheuristics which mean these methods are not simple task-specific heuristics, there is something 'more'. The possible solutions of a CO problem define a solution space. Metaheuristics provide dimension reduction (multilevel) or scouting techniques in the solution space (gradient, variable neighborhood search, simulated annealing, genetic algorithm, etc.) which description is task-independent [R7, R8]. In the case of graph partitioning, the dimension reduction methods are the most effective [R9], while for the matrix bandwidth minimization task-specific heuristics are used [R10, R11].

The second part of this dissertation makes an attempt to find new possibilities in dimension reduction, not limited to the graph partitioning problem.

Methods and tools

Most of the experiments were implemented in own-written C++ environments which were created in Microsoft Visual Studio 2008/2012 development tools. Matlab 2008 was also applied for developing test scripts. Graphical User Interfaces were developed with Windows Forms technology. Standard file formats (.msh) were used for data transfers between the own-written and third party software elements. Gmsh [R12], an open-source mesh generation and partitioner tool was used for mesh generation and visualization of the results. I chose METIS [R13] and SCOTCH [R14] for deeper analysis from the available mesh partitioning tools METIS, CHACO, SCOTCH, JOSTLE, and PARTY [R15]. I used METIS as reference solver in performance comparisons.

A dataflow architecture for unstructured mesh computations was analysed on an Alpha-Data ADM-XRC-6T1 reconfigurable development system. Further memory interface tests were made on a Xilinx Zedboard developer board, which consists a Xilinx Zinq SoC processor. This environment has an embedded FPGA, which was programmed with the Vivado HLS 14.4 toolchain.

The performance comparison of the generalized method for applicable partial solution generation was evaluated on two scientific benchmarks TSPlib [R16] and SOPlib [R17]. DPSO [R18] was used as reference solver which is one of the best available metaheuristic solutions for SOP.

During my research, I made most of the time experimental analysis but for some special cases, I have proved some simple theorems too.

2.1. New scientific results

1. Thesis Group: Joint handling of memory access and inter-processor communication in mesh partitioning.

Related publications: [J1, C1, C2, C3, C4]

The available mesh partitioning tools do not consider data locality of the output submeshes directly, however, this is crucial in memory access optimization. Partitioning packages provide data locality improvement inside the submeshes. The achievable data locality is defined by the submeshes thus partitioning has major effect on data locality.

My goal is to create a partitioning scheme in which a limit can be defined on data locality which limit depends on the physical parameters of a processor architecture (on-chip memory size). This requirement appeared in the FPGA realizations of Dataflow Machines. These architectures were found to be the most power-efficient in the case of explicit numerical approximation of partial differential equations [R19, J1]. The success of these processors is based on the perfect caching mechanism which works with zero cache-miss. However, the input data stream can not exceed a data locality limit. Partitions which are optimized for inter-processor communication often have to be modified to fit the data locality limit, which modifications degrade the quality of inter-processor communication. The connections between the two optimization goals become important to analyze.

1.1. I showed that the minimization of inter-processor communication and the maximization of data locality are conflicting goals, thus, data locality also must be considered in mesh partitioning. I have experimentally proved that with minimal inter-processor communication growth (<1%), data locality can be increased significantly (>30%).

The optimal ordering of the mesh is necessary for the determination of graph bandwidth which is the indicator of data locality. The corresponding problem is known to be NP-hard, so there is no efficient way to generate the optimal ordering. Empirical and theoretical investigations gave some relations which make the comparison of graph bandwidth of graphs possible. The basic connections were already known, for instance, the connection between the graph diameter and graph bandwidth on which the CM [R10] and GPS [R11] methods are grounded. On the side of inter-processor communication, it was known that the

good partitions consist sphere-like submeshes because this geometry has the smallest surface (edge-cut). I have just connected the results of the two optimization problems.

Data locality can be improved optionally with the increase of inter-processor communication. If we have more processors than the chromatic number of the graph, and the graph is partitioned according to the color classes, all data dependencies become inter-processor communication. Conversely, the inter-processor communication can not be decreased under a limit (minimal edge-cut partitioning) with more processors or with worse data locality. Here, the extreme case is when only one processor takes part in the computation, and there is no inter-processor communication. Experimental results showed that partitioning has a major impact on data locality. With the same number of processors for multiple mesh instances, only 0,002 growth in the communication to computation ratio was enough to increase data locality by 30-40%.

1.2. I experimentally showed that if the boundary node set of the mesh is known, the deepest levels of the breadth-first search tree started from this set defines a critical region which has to be cut by a separator to reach better data locality. I also gave a partitioning method which cuts these regions and reaches 30-40% better data locality than METIS in the case of bipartitioning at the expense of inter-processor communication.

In many applications like spatial meshes of PDE approximations, the boundary node set is known (boundary conditions). If a breadth-first search is started from this node set, the search tree defines a depth-level structure (DLS), in which the deepest levels contain nodes which have the largest distance from the surface. DLS structure shows the regions where the center points of the largest spheres reside. The best separators for data locality have to cut these spheres.

The measurement of data locality was performed by the GPS method which is a fast and efficient mesh reordering heuristic.

1.3. I gave an extension of the graph partitioning problem (Bandwidth-Limited Partitioning) which consists the joint handling of inter-processor communication and data locality and also optimizes the number of utilized processors. I proved that the solutions of BLP are better for the FPGA-based dataflow architectures than the solutions of the original problem.

In our case, the goal of graph partitioning is to define a distribution of tasks defined by the graph which distribution leads to the fastest running time. For dataflow machines beyond uniform size and minimal edge-cut, more factors have to be considered.

1. Definition (Bandwidth-Limited Partitioning). *Given a graph $G(V, E)$, with vertex set V ($|V| = n$) and edge set E. BW_Bound, $COMM\_Bound$ and K are given parameters. A partition $Q = \{P_1, P_2, .., P_k\}$ is needed which maximizes the number of parts k considering the following conditions:*
Let $Out(P_i)$ denotes the set of outgoing edges of P_i.

$$k \leq K \tag{2.1}$$

$$\max_i \left\{ \frac{|Out(P_i)|}{|P_i|} \right\} \leq COMM\_Bound \tag{2.2}$$

$$\max_i \left\{ 2 \cdot B_{f_i}(P_i) + 1 \right\} \leq BW\_Bound \tag{2.3}$$

$$|P_i| \approx \frac{n}{k} \quad \forall i \tag{2.4}$$

Bounds on inter-processor communication (2.2) and data locality (2.3) provide the desired efficiency. Size balance is described by equation (2.4).

Because of the constrained nature of BLP, it is possible that there is no solution. In the case when BLP has no solution, one of the bounds must be relaxed to a higher value.

The limit on communication to computation ratio $COMM\_R$ defines the point where the importance of inter-processor communication and data locality are equal. $COMM\_R$ also limits the number of processors, because the communication to computation ratio becomes larger with more submeshes.

1.4. I developed a Gibbs Pole Stockmeyer (GPS)-based algorithm (AM1) which creates bandwidth-limited partitioning without optimization of the processor number (partial BLP).

The method extends the GPS reordering mechanism with a proper graph bandwidth estimation. If the bandwidth need of the already labeled part of the mesh reaches the graph

bandwidth limit, the algorithm initiates a new part until the whole mesh is partitioned [J1].

1.5. I gave a near-optimal complete BLP solution for all rectangular 2-3D structured grids with grid-type BLP partitioning.

The experiences with rectangular structured meshes gave deeper understanding of the BLP task.

Grid-type partition of an $a \times b$ mesh is a $g_a \times g_b$ grid of uniform $\frac{a}{g_a} \times \frac{b}{g_b}$ submeshes. In the case of grid-type partitioning the inequalities of BLP take a verifiable form (2.5), where $s_a = \left\lfloor \frac{a}{g_a} \right\rfloor$, $s_b = \left\lfloor \frac{b}{g_b} \right\rfloor$ and $S_a = \left\lceil \frac{a}{g_a} \right\rceil$, $S_b = \left\lceil \frac{b}{g_b} \right\rceil$ denote the possible sizes of the sides. In $|Out(P_i)|$ only the non-boundary sides are accounted, each side is multiplied by $\{0, 1, 2\}$. Let m_a, m_b denote the maximum side multipliers.

$$
\begin{aligned}
2 \cdot \min\{S_a, S_b\} + 1 &\leq BW\_Bound \\
\frac{m_a \cdot s_a + m_b \cdot s_b}{s_a \cdot s_b} &\leq COMM\_Bound
\end{aligned}
\tag{2.5}
$$

Because **ga x gb** has to be smaller than **K**, the number of possible partitions is defined by (2.6) which is small enough to use exhaustive search to find the best grid-type BLP partition. In 3D the method is still viable, because there **ga x gb x gc** has to be smaller than **K** which gives only a harmonic multiplier to (2.6).

$$
K - \left\lfloor \frac{K}{2} \right\rfloor + \sum_{i=1}^{\lfloor K/2 \rfloor} \left\lfloor \frac{K}{i} \right\rfloor
\tag{2.6}
$$

The best grid-type solution is not necessarily optimal. There are counter examples, where **ga** is not the divisor of **a** or **gb** is not the divisor of **b**. In these cases, there is a better BLP partition which is not grid-type.

1.6. I developed a METIS-AM1 hybrid method for unstructured mesh partitioning.

Since the inter-processor communication can not be decreased except with reduced number of processors, it is not surprising that the bound on it limits the number of utilized processors (**k**). Many decades of research has been made to handle inter-processor communication. If a k-way METIS partition does not fit the communication bound, it is

not probable that there is a k+1-way partition which fulfills this requirement. The base idea of the METIS-AM1 hybrid is to use METIS to define **k**. Recursive bisections are performed until the bound is not reached, and between the last two steps interval halving determinates the largest **k** for which the k-way METIS partition fits the inter-processor communication bound. For each part in the k-way METIS partition, AM1 is applied.

2. Thesis Group: Applicable Partial Solution Generation for Fast-response Combinatorial Optimization.

Related publications: [C5]

Solution time of a combinatorial optimization task mainly depends on the number of decision variables, in other words, it depends on the dimension of the solution space.

Applicable partial solution generation (APSG) is a possible way for dimension reduction. A partial solution is created as partial output in constrained time. While the partial solution is applied, the optimization of the remaining part is continued. The time need of partial solution generation defines response time of the solver because the utilization of the solution can be started. Because of the $\mathcal{NP}$-hard nature of many CO problems, the most efficient solvers are heuristics which search for solutions in an exponentially growing solution space. The quality of solutions depends on the available time for the search, thus, the response time of the optimizer and solution quality are conflicting. Applicable partial solution generation provides a better trade-off between optimization time and quality because it makes possible to restrict response time only for a subset of decision variables instead of the termination of the whole optimization process when the required response time is reached. Because multi-level methods could provide fast partitioning, the results of this thesis group are not applied to the acceleration of BLP. However, it is useful for other CO applications.

2.1 I gave a metaheuristic framework for applicable partial solution generation (Variable Subset Merger - VSM). I showed the advantages and weak points of VSM on the Sequential Ordering Poblem (SOP), the Disk Scheduling Problem (DSP) and the Generalized Assignment Problem (GAP).

I gave the basic subordinate heuristics and a hybridization method which improves the solutions of real-time algorithms.

Variable Subset Merger (VSM) describes the extension of multiple subset solutions until they form a complete solution. During the VSM process, each subset has one free decision variable and has fixed assignments for the rest of variables in it.

VSM can be used for optimization problems, where a partial solution has utilizable meaning, and response time of the optimizer is important. If the existence of a feasible solution cannot be efficiently checked (GAP example), the constraints must be seen as soft constraints. Otherwise, validation procedures can be defined to ensure feasibility (SOP example). The compulsory validation of an intermediate partial solution is the only factor that limits the usage of applicable partial solution generation, and this limitation is independent of VSM.

The results on SOPLIB benchmark indicate that VSM can support real-time handling of large combinatorial optimization problems.

The metaheuristic formulation makes the hybridization simple with the best-known real-time and not real-time methods. The solutions of best real-time heuristics can be further optimized with the same response time, and VSM makes the use of not real-time heuristics possible in real-time systems.

2.2 I experimentally proved that the possibility of subset selection improves the solutions of the subproblem solving heuristic. VSM-Ploss subproblem selection outperforms the constant selection with 7.5% on TSPLIB (n=71..280) and 26% on SOPLIB (n=200..700) respectively.

Measurements on SOPLIB instances confirm that VSM-Ploss selection (VSM-SOP) can provide better quality solutions than the VSM-Const. VSM-SOP provides 26% better solutions on these larger instances. In the VSM concept, I want to define good strategies for subproblem selection and decisions, which do not lead to bad quality solutions. Results of VSM-SOP shows that it is possible to create more efficient heuristics for the generation of fixed partial solutions by using VSM.

2.3 I experimentally proved that VSM-SOP is the most efficient solver if SOP represents minimal time task scheduling and the running time of the

optimizer is also counted in the cost. For the SOPLIB bechmark VSM-SOP outperforms DPSO if the average time need of a task is less than 1-20 seconds. VSM is presented to give applicable partial solutions fast in a hybrid optimizer. However, there are situations where VSM can provide best solutions alone. If the objective function is time, the optimizer running time becomes an additional cost. For the largest SOPLIB instance with 700 variables, VSM-SOP provides a solution with only 5% cost difference from the best available solution obtained in 364 seconds by DPSO. Assuming a 700 seconds solution with DPSO, it results in a 735 seconds solution with VSM-SOP, while the response time is 320 seconds faster, thus, the overall time of operations with optimization would be 285 seconds faster. It means more than 25% speedup and nine times faster response.

$$(DPSO\_resp1 - VSM\_SOP\_resp1) = (VSM\_SOP - DPSO) \cdot C$$
$$oper\_avg = \frac{1}{n} \cdot C \cdot DPSO$$

(2.7)

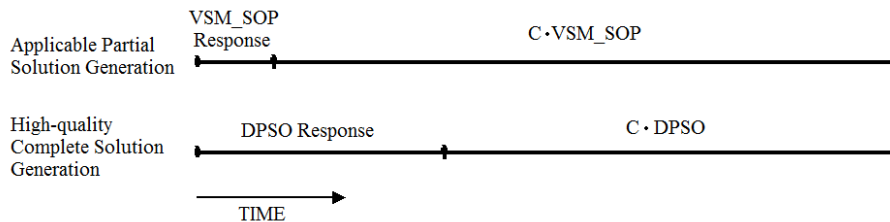Assume that SOPLIB instances describe time minimization problems. Let C denote

Figure 2.1. Comparison of operations with optimization performance on SOPLIB instances. Time exchange factor C is defined at the point where the two approaches have similar overall time need.

the exchange factor between the costs of SOPLIB solutions and execution times of the corresponding rountrips[1]. In Eq. (2.7) C is given at the point where operations with optimization cost is equal for VSM-SOP and DPSO (Fig. 2.1). With this C the average operation time cost can be obtained. If the average operation time cost is smaller than $oper\_avg$ in Eq. (2.7), VSM-SOP provides better operations with optimization cost.

---

[1]SOP is an asymmetric TSP with precedence constraints

Applicability of the results

Results of the first thesis group support the usage of dataflow machines in mesh computing. The AM1 algorithm provides access patterns with constrained data locality. The optimized and bounded access patterns are essential for dataflow machines and enables them to handle larger meshes. AM1 improves the applicability of 1-chip dataflow machines. The second part of the first thesis group provides techniques to create data locality bounded mesh partitioning. Multi-chip dataflow architectures were known for structured grids, but the definition of the corresponding partitioning problem (BLP) and solvers for the unstructured case were not given earlier.

BLP partitioning is essential for dataflow machines but has an impact on other architectures too when a submesh that is given for one chip is large enough (>300k nodes). For small submeshes, the minimization of inter-processor communication is more important than data locality. However, processor chips have more and more processing capability and off-chip DRAM, which trend makes BLP possibly important for other architectures as well. The results of the first thesis group could also be used for the determination of optimal processor number before partitioning which optimization evades the wasting of resources.

The second thesis group gives methods for response time reduction with applicable partial solution generation in combinatorial optimization. It is useful for CO problems, where a partial solution has utilizable meaning, and response time of the optimizer is important. The metaheuristic formulation makes the hybridization easy with the best-known real-time and not real-time methods. The solutions of best real-time heuristics can be further optimized with the same response time, and VSM makes the use of not real-time heuristics possible in real-time systems. The method without hybridization has been found to

be effective for task scheduling when hundreds of short (1-20 sec) tasks with precedence constraints are given.

References

Author's journal publications

[J1]   Nagy, Z. Nemes, C. **Hiba, A.** Csík, Á. Kiss, A. Ruszinkó, M. Szolgay, P. "Accelerating unstructured finite volume computations on field-programmable gate arrays". In: *Concurrency and Computation: Practice and Experience* 26.3 (2014), pp. 615–643.

[J2]   Zsedrovits, T. Bauer, P. **Hiba, A.** Nemeth, M. Pencz, B. J. M. Zarandy, A. Vanek, B. Bokor, J. "Performance Analysis of Camera Rotation Estimation Algorithms in Multi-Sensor Fusion for Unmanned Aircraft Attitude Estimation". In: *Journal of Intelligent & Robotic Systems* (2016), pp. 1–19.

[J3]   Zsedrovits, T. Bauer, P. Pencz, B. J. M. **Hiba, A.** Gozse, I. Kisantal, M. Nemeth, M. Nagy, Z. Vanek, B. Zarandy, A. Bokor, J. "Onboard Visual Sense and Avoid System for Small Aircraft". In: *IEEE Aerospace and Electronic Systems Magazine (accepted)* (2016).

Author's conference publications

[C1]   **Hiba, A.** Nagy, Z. Ruszinko, M. "Memory access optimization for computations on unstructured meshes". In: *Proc. 13th International Workshop on Cellular Nanoscale Networks and their Applications.* 2012.

[C2]   Nagy, Z. Nemes, C. **Hiba, A.** Kiss, A. Csík, Á. Szolgay, P. "FPGA based acceleration of computational fluid flow simulation on unstructured mesh geometry". In: *Field Programmable Logic and Applications (FPL), 2012 22nd International Conference on.* IEEE. 2012, pp. 128–135.

[C3]   Nagy, Z. Nemes, C. **Hiba, A.** Kiss, A. Csík, Á. Szolgay, P. "Accelerating Unstructured Finite Volume Solution of 2-D Euler Equations on FPGAs". In: *Conference on Modelling Fluid Flow (CMFF'12)*. 2012.

[C4]   **Hiba, A.** Nagy, Z. Ruszinkó, M. Szolgay, P. "Data locality-based mesh partitioning methods for dataflow machines". In: *14th International Workshop on Cellular Nanoscale Networks and their Applications*. IEEE, 2014.

[C5]   **Hiba, A.** Ruszinko, M. "Real-time combinatorial optimization with applicable partial solution generation". In: *1st International Conference on Engineering and Applied Sciences Optimization*. 2014, pp. 590–599.

[C6]   Bauer, P. **Hiba, A.** Vanek, B. Zarandy, A. Bokor, J. "Monocular Image-based Time to Collision and Closest Point of Approach Estimation". In: *24th Mediterranean Conference on Control and Automation*. 2016.

[C7]   **Hiba, A.** Zsedrovits, T. Bauer, P. Zarandy, A. "Fast horizon detection for airborne visual systems". In: *2016 International Conference on Unmanned Aircraft Systems*. 2016.

[C8]   **Hiba, A.** Orzo, L. "Retina simulator challenges, image processing with a varying resolution sensor". In: *15th International Workshop on Cellular Nanoscale Networks and their Applications*. 2016.

[C9]   **Hiba, A.** Zarandy, A. Pencz, B. "Remote Aircraft Detection against Sky Background". In: *15th International Workshop on Cellular Nanoscale Networks and their Applications*. 2016.

[C10] Orzo, L. **Hiba, A.** Zarandy, A. "Deconvolution as a model of blur adaptation in the visual cortex". In: *15th International Workshop on Cellular Nanoscale Networks and their Applications*. 2016.

[C11] Zsedrovits, T. Zarandy, A. Pencz, B. **Hiba, A.** Nameth, M. Vanek, B. "Distant aircraft detection in sense-and-avoid on kilo-processor architectures". In: *Circuit Theory and Design (ECCTD), 2015 European Conference on*. IEEE. 2015, pp. 1–4.

Related publications

[R1] Wulf, W. A. McKee, S. A. "Hitting the Memory Wall: Implications of the Obvious". In: *SIGARCH Comput. Archit. News* 23.1 (Mar. 1995), pp. 20–24. ISSN: 0163-5964. DOI: `10.1145/216585.216588`. URL: `http://doi.acm.org/10.1145/216585.216588`.

[R2] Xie, Y. "Future memory and interconnect technologies". In: *Design, Automation Test in Europe Conference Exhibition (DATE), 2013*. 2013, pp. 964–969. DOI: `10.7873/DATE.2013.202`.

[R3] Huang, Y.-J. Li, J.-F. "Yield-enhancement Schemes for Multicore Processor and Memory Stacked 3D ICs". In: *ACM Trans. Embed. Comput. Syst.* 13.3s (Mar. 2014), 106:1–106:22. ISSN: 1539-9087. DOI: `10.1145/2567933`. URL: `http://doi.acm.org/10.1145/2567933`.

[R4] Borkar, S. "Thousand Core Chips: A Technology Perspective". In: *Proceedings of the 44th Annual Design Automation Conference*. DAC '07. San Diego, California: ACM, 2007, pp. 746–749. ISBN: 978-1-59593-627-1. DOI: `10.1145/1278480.1278667`. URL: `http://doi.acm.org/10.1145/1278480.1278667`.

[R5] Garey, M. R. Johnson, D. S. *Computers and Intractablility: A Guide to the Theory of NP-completeness.* W. H. Freeman, 1979. ISBN: 0-7167-1044-7.

[R6] Papadimitriou, C. H. "The NP-completeness of the bandwidth minimization problem." In: *Computing* 16 (1976), pp. 263–270.

[R7] Blum, C. Aguilera, M. Roli, A. Sampels, M. *Hybrid Metaheuristics: An Emerging Approach to Optimization.* Studies in Computational Intelligence. Springer, 2008. ISBN: 9783540782940.

[R8] Blum, C. Puchinger, J. Raidl, G. R. Roli, A. "Hybrid metaheuristics in combinatorial optimization: A survey". In: *Applied Soft Computing* 11.6 (2011), pp. 4135–4151.

[R9] Karypis, G. Kumar, V. "Multilevel k-way partitioning scheme for irregular graphs". In: *Journal of Parallel and Distributed Computing* 48.1 (1998), pp. 96–129.

[R10] Cuthill, E. McKee, J. "Reducing the bandwidth of sparse symmetric matrices". In: *Proceedings of the ACM National Conference, Association for Computing Machinery, New York*. 1969, pp. 157–172.

[R11] Gibbs, N. Poole, W. Stockmeyer, P. "An algorithm for reducing the bandwidth and profile of sparse matrix". In: *SIAM Journal on Numerical Analysis* 13.2 (1976), pp. 236–250.

[R12] Geuzaine, C. Remacle, J.-F. "Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities". In: *International Journal for Numerical Methods in Engineering* 79.11 (2009), pp. 1309–1331.

[R13] Karypis, G. Kumar, V. "A fast and high quality multilevel scheme for partitioning irregular graphs". In: *SIAM Journal on Scientific Computing* 20.1 (1998), pp. 359–392.

[R14] Pellegrini, F. "Graph partitioning based methods and tools for scientific computing". In: *Parallel computing* 23.1 (1997), pp. 153–164.

[R15] Margoules, F. *Mesh Partitioning Techniques and Domain Decomposition Methods*. Saxe-Coburg Publications, 2007. ISBN: 978-1-874672-29-6.

[R16] WEB, *TSPLIB95 SOP problem package, http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/sop/*. 2014.

[R17] WEB, *SOPLIB problem package, http://www.idsia.ch/~roberto/SOPLIB06.zip*. 2014.

[R18] Anghinolfi, D. Montemanni, R. Paolucci, M. Gambardella, L. "A hybrid particle swarm optimization approach for the sequential ordering problem". In: *Computers and Operational Research* 38 (2011), pp. 1076–1085.

[R19] Pell, O. Bower, J. Dimond, R. Mencer, O. Flynn, M. J. "Finite-Difference Wave Propagation Modeling on Special-Purpose Dataflow Machines". In: *Parallel and Distributed Systems, IEEE Transactions on* 24.5 (2013), pp. 906–915. ISSN: 1045-9219. DOI: 10.1109/TPDS.2012.198.