# Generation of Balanced Subdomain Clusters with Minimum Interface for Distributed Domain Decomposition Applications

Dimos C. Charmpis and Manolis Papadrakakis

Institute of Structural Analysis & Seismic Research, National Technical University of Athens, 9, Iroon Polytechniou Str., Zografou Campus, GR-15780 Athens, Greece

**Summary.** Balancing and dual Domain Decomposition Methods (DDMs) are used in practice on parallel computing environments with the number of generated subdomains being generally larger than the number of available processors. The present study describes partitioning concepts used to: (a) generate subdomains for such DDMs and (b) organize these subdomains into subdomain clusters, in order to assign each cluster to a processor. The discussion concerns distributed computing environments with dedicated homogeneous processors, as well as with heterogeneous and/or non-dedicated processors. The FETI method is used to obtain numerical results demonstrating the merits of the described partitioning algorithms.

## 1 Introduction

The practical use of dual DDMs on parallel computing environments with independent numbers of subdomains and processors constitutes today a basic feature of these solution approaches (Lesoinne and Pierson [1998], Farhat et al. [2000], Charmpis and Papadrakakis [2003]). The dominant objective during the mesh partitioning task is to produce subdomains with specific geometric characteristics, in order to improve the conditioning of the arising interface problem. The number of generated subdomains is in general larger than the number of available processors, since the computational performance of dual DDMs is improved both in terms of overall execution time and storage requirements by using mesh partitions with increased numbers of subdomains. This happens because increased numbers of subdomains result in smaller profiles for stiffness and preconditioning matrices. The reduced storage demand of these matrices plays its favorable role in decreasing computational times both during the factorization of the matrices and their application in the iterative solution of the interface problem.

Since dual DDMs are applied in general to partitions with $n_s > n_p$ ($n_s$ is the number of generated subdomains and $n_p$ is the number of utilized processors), an additional computational step has to be performed just after mesh partitioning, in order to appropriately organize the $n_s$ subdomains into $n_p$ subdomain clusters and then assign each cluster to one of the $n_p$ processors. This additional computational step is termed as Subdomain Cluster GENeration (SCGEN) and can be viewed as a second partitioning task.

The issues discussed in the present study are expected to apply to all dual-type and balancing DDMs. Dual-type DDMs are all methods that are either purely based on a dual substructuring formulation (i.e. all FETI versions and variants) or include a dual substructuring kernel in the framework of a DDM of another category. Regarding the balancing DDM, its basic formulation is equivalent to a primal alternative of the FETI method (Fragakis and Papadrakakis [2003]), which indicates that both dual-type and balancing DDMs are similarly affected by the number and shape of the subdomains and the generated subdomain clusters. Therefore, in the sequel of the present work the term DDMs is used to refer collectively to dual-type and balancing DDMs.

The present study overviews the partitioning concepts used to generate subdomains and subdomain clusters for DDMs. The discussion focuses on a recently proposed heuristic approach, which handles the SCGEN task as a graph partitioning optimization problem (Charmpis and Papadrakakis [2003]). The effectiveness of heuristic SCGEN is demonstrated using the FETI method on a distributed computing environment with dedicated homogeneous processors. The applicability of this SCGEN technique is extended also to the case of heterogeneous and/or non-dedicated processors.

## 2 Mesh partitioning

DDM efficiency is governed by the convergence rate of the iterative algorithm employed for the solution of the arising interface problem. Since convergence is accelerated as the subdomains' aspect ratios improve, DDM performance is sensitive to the geometric characteristics of the subdomains. Therefore, the dominant mesh partitioning criterion that must be optimized to efficiently use DDMs is to produce subdomains with aspect ratios as good as possible. The two-step mesh partitioning strategy described in detail by Farhat et al. [1995] has been followed in the present work to produce subdomains with appropriate geometric characteristics: first the fast and simple Greedy algorithm generates a reasonable mesh partition, then the non-deterministic simulated annealing optimizer post-processes the initial partition to improve mainly the aspect ratios of the subdomains.

## 3 Subdomain Cluster GENeration (SCGEN)

Two basic requirements are specified, in order to control the SCGEN process and direct it towards the generation of subdomain clusters with suitable properties. We expect the SCGEN task to produce: (a) balanced subdomain clusters, in order to avoid excessive processor idle times and (b) small interface size between subdomain clusters, in an effort to reduce communication overhead during the iterative solution of the DDM interface problem. Thus, two distinct but complementary tasks have to be performed in order to obtain an efficient parallel DDM solution:

- The dominant objective during mesh partitioning is to ensure a favorable numerical DDM behavior. This is achieved by producing subdomains with good aspect ratios, in order to accelerate the convergence of the iterative DDM solution.
- SCGEN focuses on parallel execution aspects of the DDM, since it facilitates the exploitation of the available parallel computing environment by providing the utilized processors with balanced subsets of the DDM problem and small interface size between these subsets.

The following SCGEN approaches can be used to generate equally loaded subdomain clusters:

- The simple *Linear SCGEN* approach simulates the way early FETI runs with $n_s > n_p$ were organized, since it clusters the subdomains according to their position (numbering) in the list of subdomains: the first $n_s/n_p$ subdomains are assigned to cluster 1, the next $n_s/n_p$ to cluster 2, etc.
- Evidence of the importance of the SCGEN task was given with the *Greedy SCGEN* approach (Farhat et al. [2000]), which implements a partitioning algorithm based on the Greedy domain decomposer and exploits to some extent the adjacency information between subdomains.
- The *heuristic SCGEN* approach uses graph partitioning optimization software like METIS, JOSTLE, Chaco, etc., in order to handle in a more explicit and effective way the objectives and constraints of the SCGEN task (Charmpis and Papadrakakis [2003]).

DDM performance is affected by the quality of the generated subdomain clusters. All three aforementioned SCGEN approaches assign roughly $n_s/n_p$ subdomains to each cluster resulting in reasonably balanced processor workloads. However, the Linear and Greedy algorithms often produce rather lengthy interfaces between subdomain clusters, especially when large-size problems with irregular geometry are considered. On the other hand, heuristic partitioning is capable of detecting (near-)optimum SCGEN solutions exhibiting small interface sizes and is therefore the most effective and reliable option for the SCGEN task of DDMs.

## 4 Graph representation of mesh partition

In order to be able to exploit standard graph partitioning software in the context of heuristic SCGEN, the mesh partitioning output (which defines $n_s$ subdomains with optimal aspect ratios tailored for the DDMs) must be translated into graph data. A graph consists of vertices (associated with computational tasks), which are connected through edges (representing data dependencies between vertices). Vertices and/or edges may be assigned with weights, which implicitly specify the amount of computation associated with each vertex and the communication overhead associated with each edge. Hence, the goal of standard graph partitioning algorithms is to generate subsets of a given graph in a way that the sum of the vertex-weights in each subset is as balanced as possible and the sum of the edge-weights on the interface between the subsets (usually referred to as 'edgecut') is minimized.

In the context of the SCGEN task, the obtained mesh partitioning output is translated into an undirected graph $G = G(V, E)$, in which vertices $v_i \in V$ represent subdomains and edges $e(v_i, v_j) \in E$ are associated with subdomain connectivity ($V$ and $E$ denote the sets of vertices and edges, respectively). Thus, such a graph $G$ consists of $n_s$ vertices and contains two edges for each pair of neighboring subdomains (the adjacency of two subdomains associated with vertices $v_i$ and $v_j$ is described by both edges $e(v_i, v_j)$ and $e(v_j, v_i)$). The vertices and edges of $G$ are associated with weights, which allow the transfer of more complete information from the mesh partitioning output to its graph representation. Hence, each vertex of $G$ is weighted by the total number of degrees of freedom (d.o.f.) in the corresponding subdomain, i.e. each vertex weight implicitly specifies the computing workload of the associated subdomain. Furthermore, each edge is weighted by the number of interface d.o.f. or Lagrange multipliers between the two vertices (i.e. subdomains) the edge interconnects, in order to provide $G$ with information regarding communication overheads during the DDM solution. A characteristic graph representation of a mesh partition is illustrated in Fig. 1.

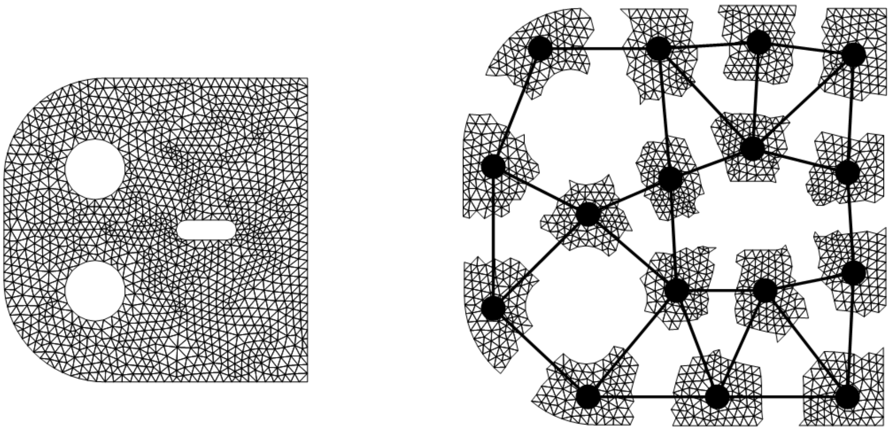

**Fig. 1.** A finite element mesh (left) and a mesh partition with its graph representation (right)

## 5 SCGEN using METIS

The construction of the mesh partition's weighted graph reduces the SCGEN task to a standard graph partitioning optimization problem. The objective of the optimization process is to minimize the interface between subdomain clusters, while the requirement of generating balanced subdomain clusters is imposed as a constraint in the partitioning process. This constraint optimization task is performed in the present work using the publicly available graph partitioning software METIS by Karypis and Kumar [1998]. Hence, the constructed weighted graph of a mesh partition is used as input to METIS, which aims in partitioning the $n_s$ vertices of the graph into $n_p$ balanced parts with minimum edgecut. Hence, METIS swaps subdomains between subdomain clusters until a locally optimal SCGEN result is reached. The output obtained contains the requested $n_p$ subdomain clusters, in which the number of d.o.f. is as balanced as possible and the number of interface d.o.f. or Lagrange multipliers between subdomain clusters is small (if not minimum).

When the available parallel computing environment consists of heterogeneous computers, each utilized processor has to be assigned with an amount of workload proportional to its processing speed. This additional requirement has to be taken into account during the SCGEN process, in order to avoid imbalanced DDM computations. A similar situation arises when other users are running a variety of jobs on the available computers resulting in a non-dedicated parallel computing environment, which suffers workloads caused by several processes and therefore utilizes processors that do not have the same processing availability. SCGEN partitions tailored for heterogeneous and/or non-dedicated processors are obtained in the present work with the use of specialized METIS routines (denoted as WMETIS), which can handle prescribed partition weights. The SCGEN solutions yielded by WMETIS heuristics ensure that the computing workload associated with each generated subdomain cluster is proportional to the processing speed or availability of the corresponding processor.

## 6 Numerical tests

The performance of the described SCGEN approaches is investigated using a six-storey building, which is modeled with a solid mesh of 11485 hexahedral 20-noded elements resulting in 249015 d.o.f. Seven mesh partitions with optimal subdomain aspect ratios are generated for this 3D building test problem. Fig. 2 illustrates the finite element mesh and a characteristic partition of the 3D building. The numerical investigation is conducted on a cluster of 12 ethernet-networked homogeneous PCs (each with a Pentium III 500 MHz processor and 256 MB RAM) using the Linux operating system and the message passing software PVM. A network-distributed one-level FETI implementation described by Charmpis and Papadrakakis [2002] and enhanced by Charmpis

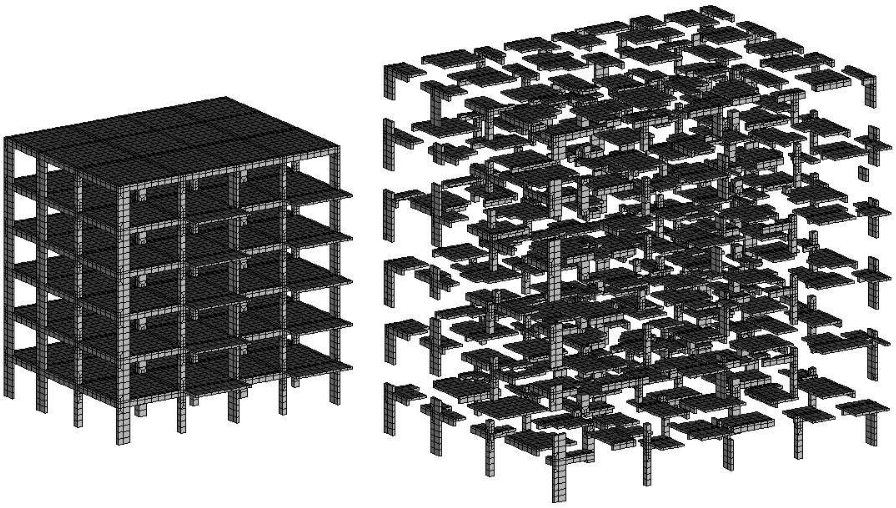

**Fig. 2.** 3D building test problem: finite element mesh (left) and mesh partition with $n_s = 250$ subdomains (right)

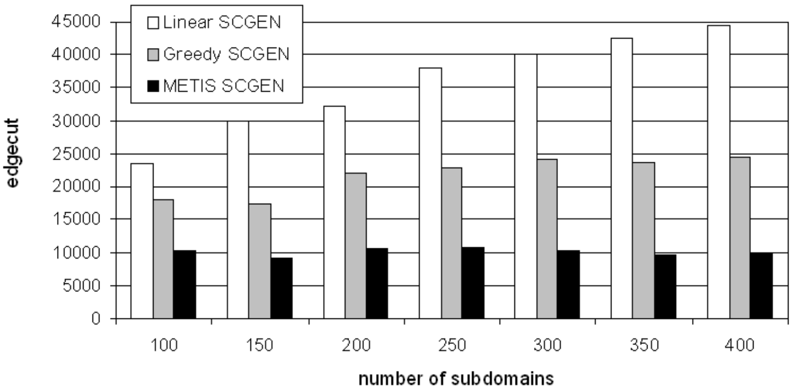

**Fig. 3.** Edgecuts obtained for several $n_s$-values ($n_p = 12$)

and Papadrakakis [2003] is executed on the PC cluster using the Dirichlet preconditioner stored in single precision arithmetic and a PCPG convergence tolerance $\epsilon = 10^{-3}$.

Fig. 3 reports edgecut-results for the 3D building test problem obtained by the SCGEN approaches on the dedicated PC cluster. According to these results, the heuristic METIS approach possesses a clear advantage over the Linear and Greedy algorithms giving by far smaller interface sizes between subdomain clusters.

Table 1 presents the time allocation for characteristic FETI runs and demonstrates the effect of low-edgecut SCGEN solutions on FETI perfor-

**Table 1.** Wall-clock time allocation of FETI runs ($n_s = 250$, $n_p = 12$)

| SCGEN algorithm | Mesh partitioning | SCGEN | Formation of FETI matrices | PCPG solution |
|---|---|---|---|---|
| Linear | 4.8s | - | 40.9s | 204.4s |
| Greedy | 4.8s | 0.05s | 41.1s | 132.9s |
| METIS | 4.8s | 0.06s | 39.2s | 69.0s |

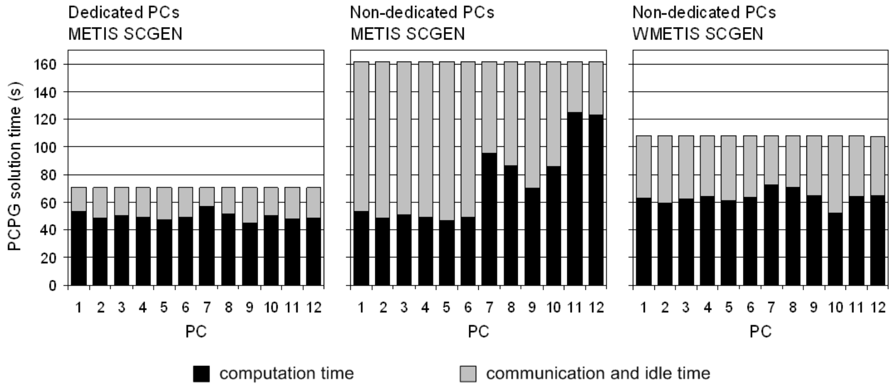

**Fig. 4.** Time allocation of the PCPG solver executed on dedicated or non-dedicated PCs with METIS or WMETIS SCGEN

mance. METIS markedly outperforms the Linear and Greedy SCGEN algorithms, since the large edgecut-values generated by the last two partitioners lead to excessive communication costs during the solution of the FETI interface problem by the iterative PCPG procedure. Furthermore, Table 1 shows that SCGEN requires a fraction of computing time compared to the overall solution effort even in the case of the METIS heuristic, therefore using parallel graph partitioning software to accelerate the SCGEN task is not necessary.

The left graph of Fig. 4 illustrates the reasonably balanced PCPG computations observed on the 12 dedicated PCs with the use of METIS SCGEN. In order to demonstrate the effectiveness of WMETIS SCGEN, a dummy computational process is started on each of PCs 7-10 and two dummy processes on each of PCs 11 and 12. This results in an artificially non-dedicated parallel computing environment, since the utilized PCs do not exhibit the same processing availability due to the waste of computing power caused by the imposed processor workloads. As shown in the middle graph of Fig. 4, METIS SCGEN cannot effectively handle the reduced processing capability of PCs 7-12 and causes severely imbalanced PCPG computations. This deficiency is alleviated by employing WMETIS SCGEN (see right graph of Fig. 4), which produces subdomain clusters using properly adjusted partition weights and is therefore capable of restoring the balancing of PCPG computations among all processors. It is noted that the communication times of Fig. 4 do not correspond only to message passing through the ethernet network, but include also the times needed to pack the communicated data into send buffers and unpack them from receive buffers.

## 7 Concluding remarks

The use of graph partitioning optimization algorithms allows the detection of (near-)optimum SCGEN solutions within large-size and irregularly structured search spaces. Such algorithms can effectively handle arbitrarily partitioned unstructured problems, while advantages over other SCGEN alternatives can be provided even for problems meshed and decomposed in a structured way (Charmpis and Papadrakakis [2003]). This consistent performance of heuristic SCGEN leads to distributed DDM runs with minimum communication overheads and reasonably balanced computations on dedicated homogeneous processors, as well as on heterogeneous and/or non-dedicated processors.

## References

D. Charmpis and M. Papadrakakis. Enhancing the performance of the FETI method with preconditioning techniques implemented on clusters of networked computers. *Computational Mechanics*, 30(1):12–28, 2002.

D. Charmpis and M. Papadrakakis. Subdomain cluster generation for domain decomposition methods using graph partitioning optimization. *Engineering Computations*, 20(8):932–963, 2003.

C. Farhat, N. Maman, and G. Brown. Mesh partitioning for implicit computations via iterative domain decomposition: impact and optimization of the subdomain aspect ratio. *International Journal for Numerical Methods in Engineering*, 38:989–1000, 1995.

C. Farhat, K. Pierson, and M. Lesoinne. The second generation FETI methods and their application to the parallel solution of large-scale linear and geometrically non-linear structural analysis problems. *Computer Methods in Applied Mechanics and Engineering*, 184:333–374, 2000.

Y. Fragakis and M. Papadrakakis. The mosaic of high performance domain decomposition methods for structural mechanics: formulation, interrelation and numerical efficiency of primal and dual methods. *Computer Methods in Applied Mechanics and Engineering*, 192:3799–3830, 2003.

G. Karypis and V. Kumar. METIS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices – Version 4.0. Technical report, Department of Computer Science, University of Minnesota, USA, 1998.

M. Lesoinne and K. Pierson. An efficient FETI implementation on distributed shared memory machines with independent numbers of subdomains and processors. *Contemporary Mathematics*, 218:318–324, 1998.