A New Distributed Optimization Approach for Solving CFD Design Problems Using Nash Game Coalition and Evolutionary Algorithms

Jyri Leskinen[1] and Jacques Périaux[12]

[1] University of Jyväskylä, Finland jyri.a.leskinen@jyu.fi
[2] CIMNE, Barcelona, Spain jperiaux@gmail.com

1 Introduction

For decades, domain decomposition methods (DDM) have provided a way of solving large-scale problems by distributing the calculation over a number of processing units. In the case of shape optimization, this has been done for each new design introduced by the optimization algorithm. This sequential process introduces a bottleneck.

Shape optimization is often done using gradient-based approaches because of their superior efficiency. Adjoint methods provide a mathematical approach of computing the gradients [4] using calculus of variations. Methods that combine the governing PDEs, their adjoints and shape parameters into one large system of equations are called *one-shot methods* [1, 6]. The optimal shape can be acquired by solving the system of equations only once. Evidently, this approach has several drawbacks. If the objective function is not unimodal, the method does not guarantee capturing the global optimal solution. Also, if the geometry changes are large, mesh deformation is no longer possible and the mesh has to be regenerated which makes this approach costly.

In this paper, a "distributed one-shot" method is introduced. It is based on ideas originating from the fields of game theory, domain decomposition, and evolutionary computing. The aim is to speed up convergence on one hand by decreasing computational time by intelligent parallelism using Nash game strategies and on the other hand by eliminating the bottleneck caused by sequential "state–costate – gradient" chain processing. The evolutionary approach allows the method to be used in global or non-smooth optimization.

1.1 Nash Games in Geometry and Domain Decomposition

Competitive Nash games were introduced by J. Nash [5]. In a competitive game the players maximize their payoff by taking into account the opponents' strategies. Nash games converge into a *Nash equilibrium*. For simplicity, let us consider a two-player

R. Bank et al. (eds.), *Domain Decomposition Methods in Science and Engineering XX*, Lecture Notes in Computational Science and Engineering 91,
DOI 10.1007/978-3-642-35275-1_72, © Springer-Verlag Berlin Heidelberg 2013

game. Let S_1 and S_2 be the sets of available strategies of Players 1 and 2 and J_1 and J_2 their payoff functions. A strategy pair $(\bar{x}_1, \bar{x}_2) \in (S_1, S_2)$ is a Nash equilibrium if and only if

$$\begin{aligned} J_1(\bar{x}_1, \bar{x}_2) &= \inf_{x_1 \in S_1} J_1(x_1, \bar{x}_2) \\ J_2(\bar{x}_1, \bar{x}_2) &= \inf_{x_2 \in S_2} J_2(\bar{x}_1, x_2) \end{aligned} \tag{1}$$

The above definition can be easily generalized to a Nash game with N players.

Nash games can also be applied to single-objective optimization. If the objective function J is additively separable, i.e. $J(\mathbf{x}) = \sum_{i=1}^{N} J_i(\mathbf{x}_i)$ and $\min_{\mathbf{x}} J(\mathbf{x}) = \min_{\mathbf{x_i}} \sum_{i=1}^{N} J_i(\mathbf{x}_i) = \mathbf{0}$, a *"virtual" Nash game* can be formed [3]. Since there are no true conflicts between the criteria, the global Nash equilibrium is located at the global optimum.

The Nash approach is well suited for inverse problems. The geometry can often be decomposed into smaller subgeometries which can be optimized concurrently [11]. Similarly, a domain decomposition problem for solving a partial differential equation can be considered as an inverse problem with a Nash game approach where the objective function is to minimize the discrepancy between the local overlapped subdomain solutions,

$$\begin{aligned} JF_1(g_1, \bar{g}_2) &= \int_{\Omega_{1,2}} \left| \varphi_1(g_1, \bar{g}_2) - \varphi_2(g_1, \bar{g}_2) \right|^2 \\ JF_2(\bar{g}_1, g_2) &= \int_{\Omega_{1,2}} \left| \varphi_2(\bar{g}_1, g_2) - \varphi_1(\bar{g}_1, g_2) \right|^2 \end{aligned} \tag{2}$$

where $|\cdot|$ is the L^2 norm, φ_i is the solution in the subdomain Ω_i and g_i is the vector of values of φ_i on the subdomain interface boundary $\Gamma_{i,j}$. $\Omega_{1,2}$ is the overlapping region (cf. Fig. 1).

In [3, 7], a hierarchical leader–follower Stackelberg game consisting of a pair of Nash games was implemented for nozzle shape reconstruction. The shape players reconstructed the target geometry using a "leader" Nash game, and the flow players reconstructed the flow using a "follower" Nash game. For each new geometry candidate produced by the shape players, a Nash game was run between the flow players. In this paper, a new Nash evolutionary approach is introduced. It replaces the computationally expensive hierarchical game by a single parallel global Nash game coalition.

1.2 Global Nash Game Coalition Algorithm (GNGCA)

The proposed method operates as follows. The geometry of the configuration is divided into subgeometries allocated to shape players whose task is to optimize the shape (or reconstruct the target geometry). Similarly, the flow players minimize the deviation of local solutions on the overlapped region of subdomains. Each shape and flow player evaluate deviation of local solutions or shape optimization with his own Evolutionary Algorithm (EA). After some frequency period, for example a single generation, shape and flow players exchange the elite values among each other. This means the flow is reconstructed along with the geometry making this a "distributed one-shot" method.

This new method is inherently parallel and therefore especially suitable for distributed parallel environments. At the higher level, the flow and shape players operate separately. Depending on the methods used, the optimization process can also be distributed. If an optimizer is used in flow reconstruction, it too can be parallelized. By reducing dimensionality of the geometry problem, algorithmic convergence can be significantly improved. For example, in the case of multi-modal problems splitting the territory can reduce the number of local optima. However, the efficiency of virtual Nash approach is highly dependent on the selected geometry decomposition. Non-optimal splitting can lead in reduced efficiency of the algorithm [11].

2 Test Case Description

The method is validated using a simple position reconstruction problem from the field of computational fluid dynamics. The geometry of the problem consists of a large disk element (radius $\frac{1}{2}$ units) surrounded by $N \geq 2$ smaller disk elements (radii $\frac{1}{8}$ units). The smaller elements are allowed to move in an area constrained by the number of elements: using radial coordinates, $r_k = 2.0^{+0.5}_{-1.3675}$ and $\theta_k = -k\frac{2\pi}{N} - \frac{\pi}{N} \pm \frac{\pi}{4N}$ (see Fig. 1).

This geometry allows the study of a wide variety of different domain and geometry decompositions (cf. Fig. 1 for a 3 element case). The test case can be made more challenging for example by deforming the shapes of the elements. In this paper, 2 and 6 element cases were studied.

The flow is described by the steady compressible potential flow,

$$\begin{aligned} \nabla \cdot \rho \nabla \varphi_k &= 0 \quad \text{in } \Omega_k \\ \varphi_k &= \mathbf{v}_\infty \text{ on } \Gamma_\infty \\ \frac{\partial \varphi_k}{\partial \mathrm{n}} &= 0 \quad \text{on } \Gamma_{1,\dots,n} \\ \varphi_k &= \varphi_j \text{ on } \Gamma_j \\ \varphi_k &= \varphi_\ell \text{ on } \Gamma_\ell \end{aligned} \tag{3}$$

where k is the index of the subdomain, and j, ℓ the right and left side neighbor domain indexes. Free-flow velocity $\mathbf{v}_\infty = (v_x, v_y) = (v_\infty \cos\alpha, v_\infty \sin\alpha)$, $|v_\infty| = 1$. The angle of attack $\alpha = 0.0°$. The density ρ is calculated using the formula $\rho = \left\{1 + \frac{\gamma-1}{2} M_\infty^2 \left(1 - |v|^2\right)\right\}^{\frac{1}{\gamma-1}}$. The constant $\gamma = 1.4$ is the ratio of specific heats for air. With a free flow Mach number $M_\infty = .3$ the flow is subsonic in the whole domain.

The objective is to reconstruct the original positions of the elements by minimizing the L^2 norm of pressure difference between the computed and target surface pressures: $JS_k(\mathbf{x}_k) = \frac{1}{n_{p_k}} \sum_{i=1}^{n_{p_k}} \left| p_{k_i} - p_{k_i}^{target} \right|^2$ where $\mathbf{x}_k = (r_k, \theta_k)$ is the decomposed design vector and n_{p_k} is the number of pressure points in the region of the decomposed geometry. The vector p_k includes the relevant surface pressure values. The global objective function is the sum of local functions.The objective function for the flow players is the L^2 norm of the discrepancy on the overlapped subregion (Eq. 2).

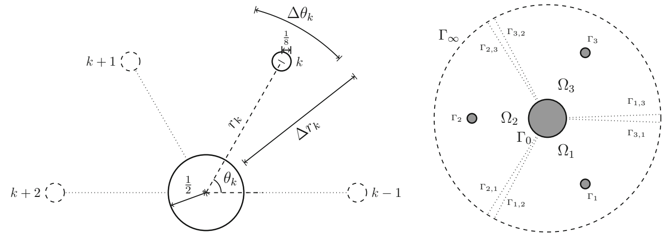

Fig. 1. Test case geometry and example decomposition

3 Test Setting

A variant of the popular Differential Evolution (DE) algorithm is used as the optimization platform. The algorithm, differential evolution with adaptive control parameters (jDE) is described in detail in the original paper [2]. The difference compared to the standard differential evolution is that the two control parameters, mutation factor F and crossover rate CR are not kept fixed. Instead, each member of the population has individual values which are allowed to change between given ranges. When a new individual is formed, the offspring inherits the values from its progenitor, or new random values are generated with probability of τ_1 for F and τ_2 for CR. In this work the population size $NP = 10n_{dim}$ was used where n_{dim} is the number of dimensions in the decomposed design vector, i.e. each instance of algorithm uses an equal number of individuals in order to make comparing them fair. Mutation factor is allowed to vary within the range $F = [0.1, 1.0]$ and crossover rate $CR = [0.0, 1.0]$. The control parameter replacement probabilities are set to $\tau_{1,2} = 0.1$. The algorithms end when the stopping criteria $JS_k = 10^{-5}$ is reached.

Because the algorithms work in parallel, a generational approach would cause bottlenecks because of the non-constant fitness function computation times. Instead, a non-generational approach is used where the older individuals are replaced immediately if the offspring is superior. In addition, the elite information exchange is done asynchronously.

Three different approaches are tested. In the first one, the jDE algorithm is run traditionally using full domain and design vector. For the second approach, a "geometry decomposition" approach introduced in [9] is used ("Nash-jDE"). The design vector $\mathbf{x} = (r_1, \theta_1, \ldots, r_N, \theta_N)$ is divided between the elements $(\mathbf{x}_k = (r_k, \theta_k), k = \{1, \ldots, N\})$, which are then optimized using several jDE algorithms operating on separate subpopulations. After each generation, the global design vector is updated using elite values from each subpopulation. The proposed GNGCA algorithm is used in the third case. For flow reconstruction, since the flow is subsonic, the additive Schwarz domain decomposition algorithm is sufficient. The overlapped regions of subdomains are made of one strip. The computational domain is divided radially so that each subdomain contains one element (Fig. 1).

The FreeFEM++ v3.18 software is used as the solver [8]. The flow is computed using finite element method with a preconditioned conjugate gradient algorithm.

Table 1. Performance of the algorithms. The symbol n_{sl} refers to the number of (shape player) slave processes, t is the wall-clock time in seconds and n_{it} to the number of objective function evaluations required by the algorithm in order to reach the target precision.

		jDE		Nash-jDE		GNGCA		speed-up	
case	n_{sl}	t	n_{it}	t	n_{it}	t	n_{it}	jDE	N-jDE
2 elements	2	1155.00s	815	390.83s	279	306.57s	514	3.77×	1.27×
	4	332.05s	474	210.97s	302	194.74s	652	1.70×	1.08×
	6	190.42s	412	132.62s	279	174.60s	888	1.09×	0.76×
6 elements	6	3632.85s	4387	971.17s	1175	171.61s	1894	21.17×	5.66×
	12	1742.23s	4226	333.90s	809	115.87s	2502	15.04×	2.88×
	18	1201.11s	4369	244.53s	880	114.08s	3743	10.53×	2.14×

Since the flow is nonlinear, Eq. 3 is solved iteratively until the threshold value of $\varepsilon_\rho = 10^{-10}$ for density is reached. The algorithms are run on a computer containing 64 Intel Xeon CPU cores clocked at 2.67 GHz.

The mesh is constructed using Triangle v1.6 Delaunay mesh generator [10]. Numerical noise is minimized using mesh regeneration with the Laplacian. In order to avoid inverse elements and maintain mesh quality, the mesh is regenerated over certain intervals ($\delta r_k = 0.1$, $\delta\theta_k = 10°$). An example decomposed mesh is illustrated in Fig. 3. Computing one subdomain gives speed-ups ranging from 3.2× to 14.0×.

4 Results and Discussion

The elapsed wall-clock times and the number of objective function evaluations required by each of the algorithm are listed on Table 1. Convergence curves of the algorithms are shown in Fig. 2. Final mesh and reconstructed global pressure field are compared to the reference in Fig. 3.

The results demonstrate that the geometry decomposition method using virtual Nash games can be used to increase algorithmic efficiency in geometry reconstruction problems. The proposed global Nash game approach shows that reconstructing geometry and flow simultaneously the wall-clock time can be reduced dramatically, provided the difference in the size of global and decomposed domains is sufficiently large. In the case of six domains, the speed-up compared to the original method is massive, over 20×. The increase compared to the pure geometry decomposition approach is also notable, over 5×. If the algorithms are compared a bit more fairly, i.e. the flow players are considered equal to the shape players, the speed-ups are 10× and 2×.

The efficiency of flow reconstruction is critical for the success of the proposed algorithm. Finding the correct geometry in an incompletely reconstructed flow field is not possible, which is evident in the large number of shape player objective function iterations needed. Unlike in the case of the other methods, increasing the number of slave processes brought only limited speed-ups for GNGCA. This was due the fact

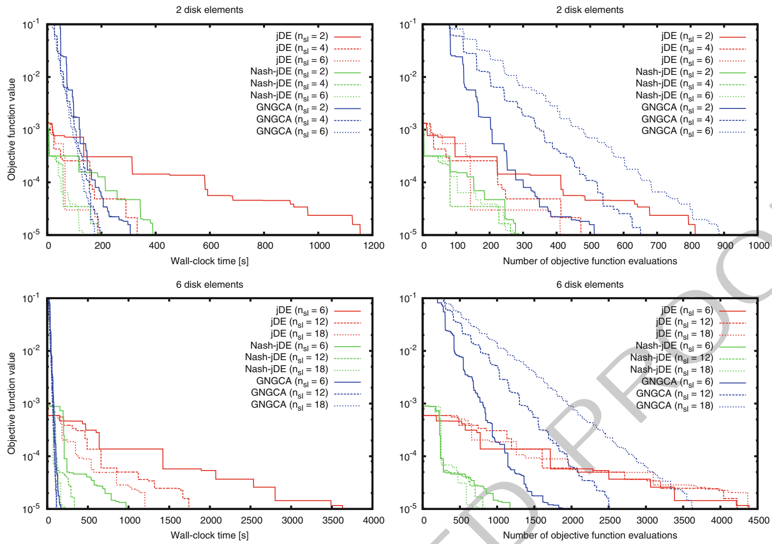

Fig. 2. Convergence curves of the tested algorithms. The onvergence according to the wall-clock time spent is on the *left* and the algorithmic convergence based on the required number of iterations is on the *right*

the flow players did not feed the shape players with accurate flow information fast enough resulting in an increased number of shape player iterations and correspondingly reduced efficiency improvement.

Algorithmic convergence can be improved by reducing the complexity of the problem. A classical method where the boundary nodes are used as shape design variables may be problematic due to a large number of variables. The situation can be improved using parallel algorithms and Bézier spline parametrization. In cases involving highly compressible potential flows where the flow is locally supersonic the domain reconstruction has to be augmented with an optimizer. The flow can be reconstructed using fast gradient methods on linearized equations coupled by DDM, or analogously to the shape presentation, the number of variables on interface boundary can be reduced using parametrization and the nonlinear flow can be reconstructed with evolutionary algorithms (cf. [3]).

5 Conclusion and Future

In this paper first results for a new "distributed one-shot" method that applies virtual Nash games, domain and geometry decomposition methods, are presented and discussed. The feasibility of the method is validated using an academic test case consisting of position reconstruction in a subsonic nonlinear flow.

In the forthcoming step, the Schwarz domain decomposition algorithm will be replaced with more robust methods. The simple compressible potential flow equa-

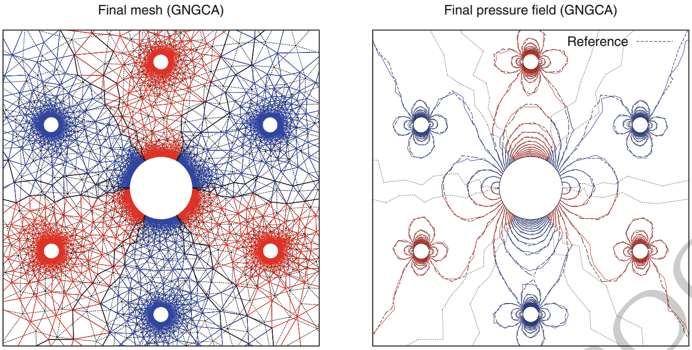

Fig. 3. Example final mesh and pressure field (GNGCA) compared to the reference

tion will be replaced with nonlinear systems of equations including Euler, Navier–Stokes, and Maxwell equations. Further tests involve complex geometries such as multi-element airfoils. The implementation of GPUs is also being studied. The ultimate target is to extend the method to speed up the capture of solutions of complex large scale problems which are frequently met in particular in 3D industrial detailed design.

Bibliography

[1] E. Arian and S. Ta'asan. Shape optimization in one shot. In *Optimal design and control*, pages 23–40. Birkhäuser Boston, Boston, MA, 1995.

[2] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*, 10(6): 646–657, December 2006.

[3] H.-Q. Chen, R. Glowinski, and J. Périaux. A domain decomposition/Nash equilibrium methodology for the solution of direct and inverse problems in fluid dynamics with evolutionary algorithms. In U. Langer et al., editor, *Domain Decomposition Methods in Science and Engineering XVIII*, Heidelberg, 2006. Springer.

[4] A. Jameson. Aerodynamic design via control theory. In *Recent advances in computational fluid dynamics*, pages 377–401. Springer, Berlin, 1989.

[5] J. F. Nash, Jr. Equilibrium points in *n*-person games. *Proc. Nat. Acad. Sci. U. S. A.*, 36:48–49, 1950. ISSN 0027–8424.

[6] E. Özkaya and N. R. Gauger. Single-step one-shot aerodynamic shape optimization. In *Optimal control of coupled systems of partial differential equations*, pages 191–204. Birkhäuser Verlag, Basel, 2009.

[7] J. Périaux, H.Q. Chen, B. Mantel, M. Sefrioui, and H.T. Sui. Combining game theory and genetic algorithms with application to DDM-nozzle optimization problems. *Finite Elem. Anal. Des.*, 37:417–429, 2001.

[8] O. Pironneau, F. Hecht, Le Hyaric A., and J. Morice. FreeFem++, www.freefem.org, 2012.

[9] M. Sefrioui and J. Périaux. Nash Genetic Algorithms: examples and applications. In *CEC00: Proceedings of the 2000 Congress on Evolutionary Computation*, pages 509–516. IEEE, November 2000.

[10] J. R. Shewchuk. Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. In M. C. Lin and D. Manocha, editors, *Applied Computational Geometry: Towards Geometric Engineering*, pages 203–222, Berlin, May 1996. Springer-Verlag.

[11] Z. Tang, J.-A. Désidéri, and J. Périaux. Distributed optimization using virtual and real game strategies for aerodynamic design. Technical Report 4543, INRIA, September 2002.