REDUCING COMPUTATION TIME IN OPTIMIZATION PROCEDURE FOR FIXTURE LAYOUT BASED ON SWARMITFIX

Xiong Li[a], Matteo Zoppi[b], Rezia Molfino[c]

[a]DIMEC, University of Genoa
[b] DIMEC, University of Genoa
[c] DIMEC, University of Genoa

[a]lixiong@dimec.unige.it, [b]zoppi@dimec.unige.it, [c]molfino@dimec.unige.it

ABSTRACT

This paper addresses a method to reduce computation time of optimization procedure for fixture layout based on a developing project self-reconfigurable swarm intelligent fixture system (SwarmItFIX) funded by European Commission FP7. The SwarmItFIX combines flexibility, self-reconfigurability, automation, and swarm multi-agent cooperation. Based on a developed optimization procedure which combines genetic algorithm with finite element analysis, the proposed method for reducing its computation time includes the simplified finite element modeling method, dynamic mutation probability and tree-database for adjusting the optimization algorithm. Simulation is finally used to prove the efficiency of the method proposed.

Keywords: SwarmItFIX, flexible manufacture, genetic algorithm, computation time reduction

1. INTRODUCTION

Towards the current trend of life-cycle design, sustainability production, and geometrical complexity, the effect on manufacture equipment is gradually shifting towards more flexibility, reconfigurability, and automation. The same tendency affects the development of fixtures, the devices used for locating and clamping the workpieces. Aiming to such trend, an EU FP7 project--self-reconfigurable swarm intelligent fixture system (SwarmItFIX), target for thin-sheet workpiece, is being developed, Molfino, Zoppi and Zlatanov (2009).

During the development, there is a problem that how to define the support position since for thin-sheet workpiece, the deformation generated by the manufacturing operations and gravity cannot be neglect. An optimization procedure combining genetic algorithm with finite element analysis was developed. However, during the simulation, we found the computation cost is time-consuming. This paper is going to address the solution for reducing the computation time.

An, Choi, and Kim (2003) proposed a hybrid algorithm to reduce the computation time of genetic algorithm. Zulkarnain (2010) adopted the method of reducing computation steps to increase the conputation efficiency. Schonning, Nayfeha, Zarda (2003) utilized the dependency-tracking language to reduce computational time during multidisciplinary design optimization.

From practical composition of the optimization procedure, by absorbing previous researchers' achievements, this paper will reduce the computation time from three aspects: to simplify finite element modeling, to modify genetic algorithm parameters, and to construct tree-type database.

The structure of this paper is like that, after having introduced the characteristics and composition of SwarmItFIX, the optimization procedure combining genetic algorithm and finite element analysis for the purpose of obtaining the optimal fixture layout for flexible workpiece is presented. Based on such optimization procedure, the method to reduce the computation time is proposed. The simulation cases of fixture layout optimization demonstrate the efficiency of this method.

2. SELF-RECONFIGURABLE SWARM INTELLIGENT FIXTURE SYSTEM

As illustrated in Figure 1, the SwarmItFIX consists of a bench and several actuated fixtures (agents), which collaborate, without human interference, to support a thin-sheet workpiece. The bench provides a surface for the agents to move, and lock once they are at the desired location. It will also incorporate the power-supply and communication systems, as well as the means to measure the accurate position of the agents.

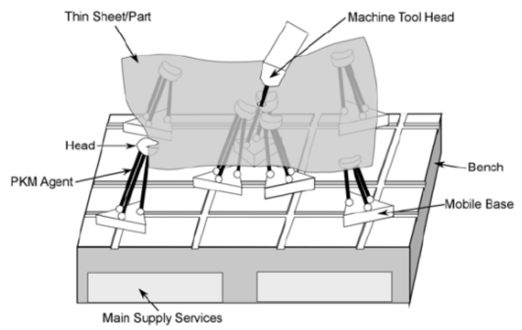


Figure 1: General Scheme Of The SwarmItFIX System

Each agent is composed of one mobile base, one parallel mechanism (PM), and one adjustable head. The mobile base supports the whole agent and hosts all the electronic parts for control. It provides the locomotion of the agent on the bench and communicates with the bench. The PM provides necessary workspace, support, and six degree of freedom for the head. The head directly contacts the sheet panel and has smart material with phase-change capabilities to conform to the local geometry of the workpiece. During the machining process, the fixtures will change positions on the bench, to form different supporting layouts as adapting to the process plan. The developed system towards prototype is shown in Figure 2.

Figure 2: The Developed SwarmItFIX

Beside its primary target application, the aerospace industry (fuselage sections, aerofoils, and other panels), in the future this system can be beneficial to other secondary sectors as shown in Figure 3.

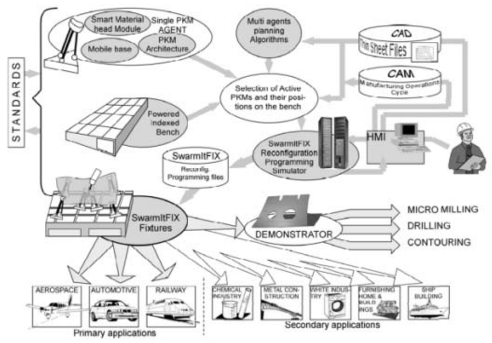


Figure 3: SwarmItFIX System Design and Development: Primary and Secondary Beneficial Sectors

3. OPTIMIZATION PROCEDURE

The optimization algorithm combines FE analysis to the use of a genetic algorithm. Genetic Algorithms (GA) based optimization is radically different from the search optimization methods, including traditional optimization methods and other stochastic methods (Renner, and Ekart 2003; Krishnakumar and Melkote 2000). No derivates is needed and it can escape local minima. Its inductive nature makes it do not need to know any rules of the problems and work by its own internal rule. Thus, GA is ideally suited for the fixture layout optimization problem since only the design variables are used with no gradient or other auxiliary information (Goldberg 1989).

The fixture layout optimization procedure developed is shown in Figure 4.

In each generation of individuals generated by the GA, the individual (support position) is sent to ABAQUS to calculate the deformation. The values are passed to the genetic algorithm as fitness values until the final iteration condition is met. The final iteration condition is the iteration number reaching the upper limit of the generation number. All the code of genetic algorithm is implemented in MATLAB, and the finite element parameter modeling is implemented with Python, the program language of ABAQUS.

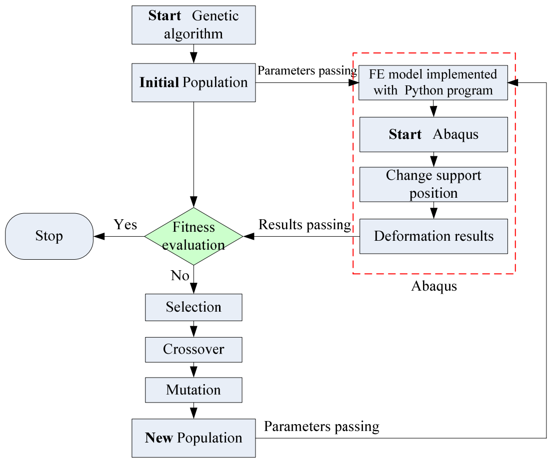


Figure 4: Optimization Procedure

The algorithm of the proposed optimization procedure is shown below.

Algorithm:

Step 1: Set the parameters of genetic algorithm

Step 2: Initialize the population, randomly generate individuals

Step 3: Convert the binary individuals into the support positions of fixtures

Step 4: Pass the converted data to FE program to change the support positions of fixtures

Step 5: Run FEA

Step 6: Return the deformation to GA program

Step 7: Use the deformation for fitness valuation

Step 8: If the fitness satisfy the convergence condition, go to Step 11.

Step 9: Follow the normal genetic algorithm procedure: selection, crossover and mutation

Step 10: Get the new population, go to Step 3

Step 11: Return the final solution and Stop

4. REDUCTION OF COMPUTATION TIME

As noticed on many simulation cases, this optimization procedure is time-consuming. There are three possibilities to reduce the computation time: to simplify FE modeling, to modify GA parameters, and to adjust the optimization procedure.

4.1. Simplification of Finite Elements Modeling

During the FE modeling, the interaction between head and workpiece needs to be set. Usually, such interaction is set as contact model, either point-to-point contact or point-to-surface contact (Satyanarayana and Melkote 2004). These contact models involve nonlinear computation with a significant increase in model complexity compared to linear modelling. However, the heads, as shown in Figure 5, apply to the workpiece a 6-DOF constrain and their stiffness is higher than the bending stiffness of the workpiece. Since heads never detach from the workpiece (what would require modeling with contacts), we can use the linear computation tie constraint of ABAQUS in place of contacts reducing remarkably the complexity of the FE model.

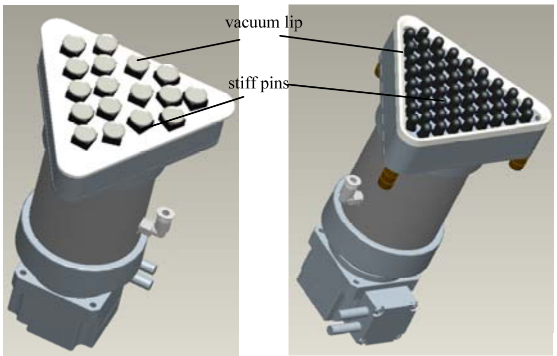


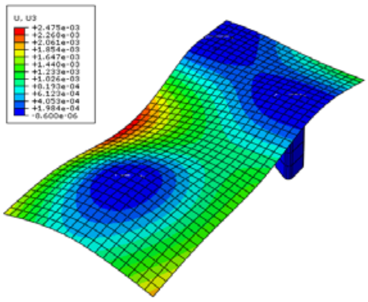
Figure 5: Two Kinds of the Developed Head

Simulation results derived from different typical cases are shown to check feasibility of such simplification as shown in Table 1. We can find that the deformation under contact model is average 7% larger than the one with tie constraint model. Therefore, such simplification is acceptable either from the point of view of practical application and computation results.

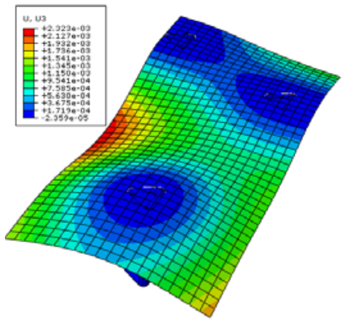
Table 1: Comparison of Simulation Results under Different Interactive Model

Workpiece dimension	Maximum deformation under different models	
	Under contact model	Under tie constraint
2D case 2m× 1m× 0.004m	2.475e-3m	2.323e-3m
3D case 0.6m× 0.7m Thickness 0.004m Average curvature 0.0011 mm^{-1}	3.25e-5m	3.12e-5m

One of the simulation cases is shown in Figure 6.



(a) Deformation under Contact Model



(b) Deformation under Tie Constraint

Figure 6: Optimization Results of 2D Cases under Different FE Modeling

4.2. Modification of Genetic Algorithm Parameters

From the genetic algorithm itself, dynamic mutation probability can be used for speeding up the convergence velocity and allowing large possibility to search global optimal solution.

A compromise has been found by trial and error between velocity of convergence and risk to fall in local minima. The experimental mutation rate obtained is:

$$p_m = p_m - \frac{j^4 - (p_m - \frac{p_c}{k})}{k^4} \tag{1}$$

Where j is the generation number, k is the iteration number, p_m is the mutation probability and it changes with the generation number until the maximum number of iterations is reached. With the increase of the number of generation, the mutation probability gradually decreases. The relationship between the GA convergence and the variable mutation rate is also consistent with the relationship in (Beasley and Chu 1996).

4.3. Adjusting the Optimization Procedure

Since this optimization procedure is based on random generation, inevitably, there will be some individuals (support positions) identical to the individuals in the previous population pool. In this case, a tree-type database, as shown in Figure 7, can be constructed to reduce the repeated computation.

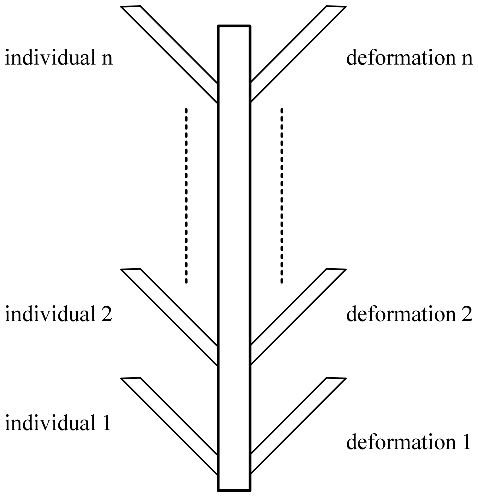


Figure 7: Tree-type Database

In this tree-type database, there is a rule of correspondence between the left branches and the right branches such that there is a deformation value assigned to each individual in the left branches. The "tree" will grow up with the increasing of iteration number. Therefore, before running FEA at the Step-5 of the algorithm, each individual will compare with the data in this tree-type database. If the individual is the same, there is no need to run FEA. It can obtain the deformation directly from the branches and thus save the computation time. Conversely, if there is no individual in the tree to match the individuals which need to be computed, FEA will be run and correspondence results will be saved in new branches. Such adjusting can save the computation time, obviously, especially during the convergence stage, since many individuals are the same with those in the previous population pool.

After all the above modification, the time-efficient optimization algorithm is illustrated as follows.

New Algorithm:

Step 1: Set the parameters of genetic algorithm and empty matrix for tree-database

Step 2: Initialize the population, randomly generate individuals

Step 3: Convert the binary individuals into the support positions of fixtures

Step 4: Pass the converted data to simplified FE program to change the support positions of fixtures

Step 5: Run FEA

Step 6: Return the deformation to GA program

Added Step 1: Keep the individuals and deformation in the tree-database

Step 7: Use the deformation for fitness valuation

Step 8: If the fitness satisfy the convergence condition, go to Step 11.

Step 9: Follow the normal genetic algorithm procedure: selection, crossover and dynamic mutation

Step 10: Get the new population, repeat the operation addressed in Step 3

Added Step 2: Compare the individuals with the individuals in the database. If some individuals is the same with individuals in the database, get the corresponding deformation, go to Step 7.

Added Step 3: keep the remained individuals, go to Step 4

Step 11: Return the final solution and Stop

5. SIMULATION OF DIFFERENT CASES AND DISCUSSION

For the purpose to check the efficiency of the solution, different 2D and 3D cases are simulated. The hardware specification is: Intel Core(TM) i5 CPU 660@3.33GHz, 3.49GB of RAM. The computation was implemented in Matlab R2009 and ABAQUS V 6.10. The genetic algorithm parameters are set as listed in Table 2.

Table 2 Genetic Algorithm Parameters

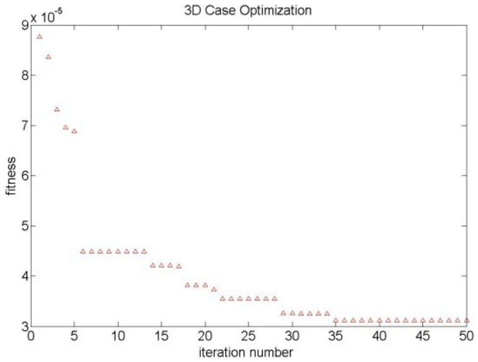
Population size	Generation number	Selection probability	Mutation probability
20	50	0.8	0.8

For the comparison of the computation time, for each case considered only the optimization algorithm is different. All the other settings and conditions are the same. Four typical cases are simulated and each of their simulation runs 10 times. Their average computation time is recorded in Table 3.

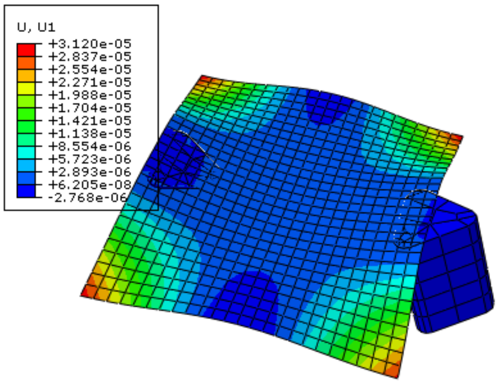
Table 3 Simulation Cases

Case type	Workpiece dimension	Fixture number	Computation time (hour)	
			Previous algorithm	New algorithm
2D	2m × 1m × 0.004m	3	175h	20h
2D	0.85m × 1.5 m × 0.004 m	3	160h	18h
3D	1m × 0.9m Thickness 0.004m Average curvature 0.0015 mm^{-1}	3	150h	14.5h
3D	0.6m × 0.7 m Thickness 0.004m Average curvature 0.0011 mm^{-1}	2	90h	10h

As shown in Table 3, the new algorithm can reduce the computation time sharply. One of the simulation case results is shown in Figure 8.



(a) Fitness Results at Each Generation



(b) Final Fitness Result Shown in FEM

Figure 12: Optimal SimulationResults of 3D Workpiece

6. CONCLUSION

A developing project about self-reconfigurable swarm intelligent fixture system which combines flexibility, self-reconfigurability, automation, and swarm multi-agent cooperation is addressed. The reduction in computation time for the development of the optimization procedure for optimal fixture layout is proposed. Such solution can be implemented in practice and has both academic and application significance.

ACKNOWLEDGMENTS

The research is developed within the SwarmItFIX project co-funded under the Seventh Framework Program (Collaborative Project 214678). The European Commission is gratefully acknowledged.

REFERENCES

An, J.W. and Choi, J.K. and Kim, N. 2003, A hybrid algorithm to reduce the computation time of genetic algorithm for designing binary phase holograms, *Journal of the optical society of Korea*, Vol.7, No. 4, pp. 264-268

Beasley J.E., Chu P.C. 1996, A genetic algorithm for the set covering problem, *European Journal of Operational Research* 94, pp. 392-404

Goldberg D.E. 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA

Krishnakumar K., Melkote S.N. 2000, Machining fixture layout optimization, *International Journal of Machine Tools and Manufacture*, 40, pp. 579–598

Molfino R., Zoppi M., Zlatanov D. 2009. Reconfigurable swarm fixtures. *ASME/IFToM International Conference on Reconfigurable Mechanisms and Robots*. Proceedings, June 22–24, 2009, pp. 730–735

Renner G., Ekart A. 2003, Genetic algorithms in computer aided design, *Computer Aided Design*, 35, pp. 709–726

Satyanarayana S., Melkote S.N. 2004, Finite element modeling of fixture-workpiece contacts: single contact modeling and experimental verification, *International Journal of Machine Tools & Manufacture* 44, pp. 903-913

Schonning M.A., Nayfeha J.F., Zarda P.R.. 2003, Utilization of a dependency-tracking language to reduce computational time during multidisciplinary design optimization, *Advances in Engineering Software*, 34, pp. 115–122

Zulkarnain Md. Ali., 2010, Reduce computation steps can increase the efficiency of computation algorithm, *Journal of Computer Science*, 6(10), 1023-1027

AUTHORS BIOGRAPHY

Xiong Li : PhD candidate in DIMEC, University of Genoa

Matteo Zoppi : PhD researcher at University of Genoa. Mechanisms and robots creative design. Involved in many European Projects and Networks (e.g. PICAV, SwarmItFIX, EURON, CLAWAR). ASME member

Rezia Molfino : Full professor at University of Genoa, responsible of PMAR lab Robotics. President of SIRI (Italian Association of Robotics and Automation), National Coordinator at IFR (International Federation of Robotics). Appointed reviewer for international journals in robotics and automation. Author and co-author of about 250 scientific papers. About 20 patents on robotic devices and systems.