A Multi-Objective Metamodel-Assisted Memetic Algorithm with Strength-based Local Refinement

Kyriakos C. Giannakoglou, Chariklia Georgopoulou

▶ **To cite this version:**

Kyriakos C. Giannakoglou, Chariklia Georgopoulou. A Multi-Objective Metamodel-Assisted Memetic Algorithm with Strength-based Local Refinement. Engineering Optimization, 2009, 41 (10), pp.909-923. 10.1080/03052150902866577 . hal-00545364

HAL Id: hal-00545364
https://hal.science/hal-00545364

Submitted on 10 Dec 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Multi-Objective Metamodel-Assisted Memetic Algorithm with Strength-based Local Refinement

Journal:	*Engineering Optimization*
Manuscript ID:	GENO-2008-0182.R3
Manuscript Type:	Review
Date Submitted by the Author:	24-Feb-2009
Complete List of Authors:	Giannakoglou, Kyriakos; National technical University of Athens Georgopoulou, Chariklia; National Technical University of Athens
Keywords:	Metamodel-Assisted Memetic Algorithms, Fitness function, Strength, Multi-objective Optimization
Note: The following files were submitted by the author for peer review, but cannot be converted to PDF. You must view these files (e.g. movies) online.	
georgopoulou.tex georgopoulou_appendix.tex georgopoulou_figures.tex georgopoulou_macros.tex georgopoulou_tables.tex georgopoulou.bib	

scholarONE™
Manuscript Central

Engineering Optimization
Vol. 00, No. 00, Month 200x, 1–18

RESEARCH ARTICLE

A Multi-Objective Metamodel-Assisted Memetic Algorithm with Strength-based Local Refinement

Chariklia A. Georgopoulou and Kyriakos C. Giannakoglou*

National Technical University of Athens, School of Mechanical Engineering, Lab. of Thermal Turbomachines, Parallel CFD & Optimization Unit, P.O. Box 64069, Athens 157 10, Greece

(*Received 00 Month 200x; final version received 00 Month 200x*)

Metamodel-Assisted Evolutionary Algorithms are low-cost optimization methods for CPU demanding problems. Memetic Algorithms combine global and local search methods, aiming at improving the quality of promising solutions. This paper proposes a Metamodel-Assisted Memetic Algorithm which combines and extends the capabilities of the aforementioned techniques. Herein, metamodels undertake a dual role: they perform a low-cost pre-evaluation of population members during the global search and the gradient-based refinement of promising solutions. This reduces significantly the number of calls to the evaluation tool and overcomes the need for computing the objective function gradients.

In multi-objective problems, the selection of individuals for refinement is based on domination and distance criteria. During refinement, a scalar strength function is maximized and this proves to be beneficial in constrained optimization. The proposed Metamodel-Assisted Memetic Algorithm employs principles of Lamarckian learning and is demonstrated on mathematical and engineering applications.

Keywords: Metamodel-Assisted Memetic Algorithms; Fitness function; Strength; Multi-objective Optimization.

1. Introduction

In the literature, a considerable number of papers appeared on the use of surrogate evaluation models or metamodels to assist global search (*GS*) methods, Shan and Wang (2005), such as the Evolutionary Algorithms (*EAs*), Madsen *et al.* (2000), Ong *et al.* (2000), Ulmer *et al.* (2004, 2003), Jin *et al.* (2000), Buche *et al.* (2005), Lim *et al.* (2008). These

*Corresponding author. Email: kgianna@central.ntua.gr

ISSN: 0305-215X print/ISSN 1029-0273 online
© 200x Taylor & Francis
DOI: 10.1080/0305215YYxxxxxxx
http://www.informaworld.com

For Peer Review Only
URL: http://mc.manuscriptcentral.com/geno Email: A.B.Templeman@liverpool.ac.uk

methods can be classified based on the way metamodels are coupled with or updated during the evolutionary search. Further discussion on the so-called Metamodel-Assisted Evolutionary Algorithms (*MAEA*s) can be found in relevant review papers and books such as in Giannakoglou (2002), Keane and Nair (2005), Jin *et al.* (2002). In *MAEA*s, the role of metamodels is to provide approximations to the fitness or cost of candidate solutions and, thus, save a great amount of evaluations which would otherwise be carried out by the CPU-demanding problem-specific tool. In Karakasis and Giannakoglou (2005), Giannakoglou *et al.* (2001), they are used to pre-evaluate the population members, in the so-called Inexact Pre-Evaluation (*IPE*) phase of each generation of the *MAEA*. *IPE* is based on metamodels trained on the fly, on a small number of already evaluated individuals in the neighborhood of each inividual (local metamodels).

Memetic Algorithms (*MA*s), Dawkins (1976), Moscato (1999), Krasnogor (2002), Ong and Keane (2004), Ong *et al.* (2006), Krasnogor and Gustafson (2002), Hart (1994), are optimization methods that combine *GS* and local search (*LS*). *MA*s profit of the abilities of *EAs* to explore the most promising regions of the design space without being trapped into local optima and the efficiency of deterministic methods to further refine promising solutions located during *GS*. However, *MAs* may become CPU demanding, since they employ calls to the evaluation tool and, likely, the one computing the gradient of the objective function.

This paper presents a Metamodel-Assisted Memetic Algorithm (*MAMA*) for multi-objective optimization (*MOO*) problems, in which locally trained metamodels approximate the objective function values during the *GS* and the gradient required for the refinement. The core *GS* tool is a *MAEA* whereas an ascent method, Nocedal and Wright (1999), using the gradient provided by the metamodels is used for the *LS*. The same set of patterns is used to train a radial basis function (*RBF*), Haykin (1999), network for each individual during the *IPE* phase of the *MAEA* and the refinement process. In contrast to the *MAEA* which, in *MOO* problems, seeks the Pareto set of non-dominated solutions, *LS* copes with the maximization of a (scalar) strength function. In the literature of *MA*s for *MOO*, it is proposed either to refine only one of the objectives at a time, Bosman and de Jong (2005), or to handle a linear combination of weighted objectives, Ishibuchi and Murata (1996), Jaszkiewicz (2002), Ishibuchi *et al.* (2003), Lim *et al.* (2008). The *MPAES* algorithm proposed in Knowles and Corne (2000) is an exception since it employs a form of Pareto ranking based on comparing individuals to an archive of non-dominated solutions.

In the present paper, a new scheme is proposed. The function to be maximized in *LS* is a strength function that takes into account domination criteria. During the refinement process, the local *RBF* networks are re-trained on the strength function values, computed by considering the current offspring, parent and elite populations, and are used to compute ascent directions. This is important when the objective functions have rugged landscapes or we are dealing with constrained problems where the objective function is penalized proportionally to the degree of constraints' violation. The outcome of the *LS* process may displace the starting point, for both genotype and phenotype, according to the Lamarckian learning rules. Relevant papers on *MAMA*s almost exclusively focus on single-objective optimization (*SOO*) problems and make use of metamodels to support either both *GS* and *LS*, Zhou *et al.* (2007a,b), or only *LS*, Liang *et al.* (1999, 2000), Hacker (2002).

The proposed *MAMA* is firstly demonstrated on mathematical benchmark cases and its performance is analyzed statistically and compared to *EAs* and *MAEAs*. Two constrained engineering applications, namely the design of a combined cycle power plant and

For Peer Review Only

a compressor cascade, are also studied. All the examined cases deal with two objectives.

2. The Proposed *MAMA*

The proposed *MAMA* was built on an existing *MAEA*, Giannakoglou (2002), for *MOO* problems with M objectives. One may clearly distinguish the proposed *MAMA* from a *MAEA* or a "conventional" *MA* by (a) the *LS* algorithm which relies on local metamodels trained on a scalar strength function to be maximized and (b) the coupling of *LS* with the core *MAEA*, where a scheme for selecting individuals for *LS* is proposed.

MAMA involves four sets of candidate solutions: the offspring set $\mathcal{P}_{\lambda,g}$, the parent set $\mathcal{P}_{\mu,g}$, the archival set $\mathcal{P}_{\alpha,g}$ (which progressively approaches the Pareto front of non-dominated solutions) and an extra set $\mathcal{P}_{LS,g}$ of individuals qualified to undergo *LS*. The second index g stands for the generation counter. Candidate solutions are denoted by $\vec{x} \in \mathcal{R}^N$ and the objective function vector is $\vec{F}(\vec{x}) \in \mathcal{R}^M$. During the first few generations ($g < g^{IPE}$, where g^{IPE} is user-defined), the proposed *MAMA* behaves as a (μ,λ) *EA* and the λ offspring are all evaluated on the problem-specific tool. The so-evaluated individuals $\vec{x}$, paired with the $\vec{F}(\vec{x})$ values, are recorded in a database $\mathcal{D}$.

Dealing with *MOO* problems, it is necessary to adopt a fitness assignment technique; through this technique a unique scalar fitness value ϕ is assigned to each population member, depending on its $\vec{F}(\vec{x})$ and dominance criteria. Herein, the selected technique was inspired by the widely used *SPEA*2 (Strength Pareto *EA*), Zitzler *et al.* (2001). Two terms contribute additively to the fitness value ϕ of any individual. The first term is the raw fitness R and the second is density D. The raw fitness R of an individual is determined by the strength S (i.e. the number of dominated individuals) of its dominators in $\mathcal{P}_{\lambda,g}$. The D value of each individual is determined by its distance from the k-th closest neighbor in the normalized objectives space, where $k = \sqrt{\lambda}$. Then the scalar $\phi = \phi(R(S), D)$ value is computed, as in Zitzler *et al.* (2001). An illustrative example of the calculation of ϕ is shown in figure 1. In the sake of clarity, when ϕ is computed using approximate objective function values $\tilde{\vec{F}}(\vec{x})$ (provided by the metamodels during the *IPE* phase, see Step 1 below), this will be denoted by $\tilde{\phi}$.

The *MAMA* steps presented below apply if $g \geq g^{IPE}$, during which $\mathcal{D}$ is non-empty ($|\mathcal{D}| = \lambda_{\mathcal{D}}$) and continues to be updated by recording all new individuals evaluated on the problem-specific tool. In figure 3, the corresponding algorithm flowchart is shown.

Step 1. [*GS*: Inexact Pre-Evaluations] For each individual $\vec{x} \in \mathcal{P}_{\lambda,g}$, its K closest $\mathcal{D}$ entries are selected to populate the corresponding training set $\mathcal{T}_K$. It is recommended that K takes on a value close to (or slightly higher than) the number N of design variables. An *RBF* network (with N input and M output units; N and M are equal to the number of design variables and the number of objectives, respectively) is trained on the objective function vectors of the $\mathcal{T}_K$ patterns and an approximate $\tilde{\vec{F}}(\vec{x})$ is obtained. After the metamodel-based evaluation (for all but the $\mathcal{P}_{\lambda,g}$ members which have been previously evaluated and can be restored from $\mathcal{D}$) and by applying dominance and strength criteria, as previously described, fitness values ϕ and $\tilde{\phi}$ are assigned to all individuals in $\mathcal{P}_{\lambda,g}$. The λ individuals are sorted according to their fitness values (irrespective of the evaluation tool used) and the top λ_e of them are selected to populate $\mathcal{P}_{e,g}$, where $\lambda_e < \lambda$ is a user-defined integer.

Step 2: [*GS*: Exact evaluations] Each $\vec{x} \in \mathcal{P}_{e,g}$ is re-evaluated using the problem-specific tool and stored in $\mathcal{D}$. The $\vec{F}(\vec{x})$ values displace the metamodel-based ones for the

For Peer Review Only

$\mathcal{P}_{e,g}$ members. In constrained optimization cases, if a candidate solution violates even a single constraint, the corresponding $\vec{F}(\vec{x})$ includes an exponential penalty factor depending on the degree of constraint violation.

Step 3. [*LS*: Preparation] A number of $\mathcal{P}_{\lambda,g}$ members for which $\vec{F}(\vec{x})$ are known, is singled out in $\mathcal{P}_{LS,g}$ ($\lambda_{LS}=|\mathcal{P}_{LS,g}|$), according to distance and dominance criteria, as it will be discussed elsewhere in this paper. The $\mathcal{P}_{LS,g}$ members are selected from $\mathcal{P}_{e,g}$ ($\subset\mathcal{P}_{\lambda,g}$) and the subset $\mathcal{P}_{\lambda,g}\cap\mathcal{D}$, so that only individuals that have been evaluated on the problem-specific tool may undergo *LS*.

Step 4. [*LS*: Refinement] The previously populated $\mathcal{P}_{LS,g}$ members are refined using a gradient-based descent method which, in contrast to conventional *MAs*, aims at maximizing their strength S. This search is subject to $2N$ inequality constraints, corresponding to the upper and lower variables' bounds. To avoid calculations on the problem-specific tool during *LS*, the gradient of S is approximated using the *RBF* network trained on the S values of the $\mathcal{T}_K$ patterns associated with each individual (as formed in Step 1). The process initiates from the genotype $\vec{x}$ of the individual for which both $\vec{F}(\vec{x})$ (from Step 2) and strength S (from Step 3) are known and ends up with a new individual $\vec{x}^*$. This should be evaluated on the problem-specific tool before being accepted or not. The new individual is archived in $\mathcal{D}$ and, in case the starting objectives vector is improved, both $\vec{x^*}$ and $\vec{F}(\vec{x^*})$ displace $\vec{x}$ and $\vec{F}(\vec{x})$ in $\mathcal{P}_{\lambda,g}$ and $\mathcal{P}_{LS,g}$ according to the Lamarckian learning rules. An example of the *LS* process is shown in figure 2.

Step 5. [*GS*: Evolution] Fitness ϕ and $\tilde{\phi}$ are updated for all individuals in $\mathcal{P}_g=\mathcal{P}_{\mu,g}\cup\mathcal{P}_{\lambda,g}\cup\mathcal{P}_{\alpha,g}$. The elitism operator E is applied to $\mathcal{P}_{e,g}$ and $\mathcal{P}_{\alpha,g}$ to update the elite set and the evolution operators (parent selection P, crossover C and mutation M) to create the new offspring population. Schematically,

$$\mathcal{P}_{a,g+1} = E(\mathcal{P}_{e,g}\cup\mathcal{P}_{\alpha,g})$$
$$\mathcal{P}_{\mu,g+1} = P(\mathcal{P}_{\mu,g}\cup\mathcal{P}_{\lambda,g})$$
$$\mathcal{P}_{\lambda,g+1} = M(C(\mathcal{P}_{\mu,g+1},\mathcal{P}_{\alpha,g+1})).$$

Step 6. [Termination] If the maximum number of evaluations (on the problem-specific tool) is reached or the set $\mathcal{P}_\alpha$ remains unchanged for a number of generations, the algorithm terminates. Otherwise, $g\leftarrow g+1$ and the evolution continues from Step 1.

As already discussed, in order to maximize the expected gain from the refinement process, $\mathcal{P}_{LS,g}$ should be carefully populated. The number of refined individuals λ_{LS} is not fixed and the user defines only its upper bound λ_{LS}^{max}. In *MOO* problems, *LS* should hopefully give new solutions that either dominate one or more members of the current archival front or fill the gap between two non-dominated members or stretch the current front. In problems with multimodal objective functions, it is evident that whether a gradient-based search may or may not achieve any of these goals depends on the starting point. Thus, the formation of $\mathcal{P}_{LS,g}$ is based on three selection rules:

- $\mathcal{P}_{LS,g}$ is populated by individuals evaluated on the problem-specific tool.
- Candidates with higher strength S must be selected with priority; S is calculated by processing $\mathcal{P}_{\lambda,g}$ without taking into consideration the evaluation tool (problem-specific tool or metamodel).
- A candidate is not allowed to enter $\mathcal{P}_{LS,g}$ if it lies in the vicinity of a previous entry.

To summarize, among neighboring candidates, the most dominant one is chosen and the

For Peer Review Only

others are excluded. For instance, consider that among the two neighboring candidate solutions A and B shown with circles in figure 4, only one must be selected for *LS*. Though A does not dominate B and vice-versa, A is selected for *LS* since its strength is higher.

3. Method Application & Assessment

3.1. *Validation on Mathematical Problems*

Before going through the two engineering applications, a thorough validation of the proposed method on three two-objective function minimization problems is presented in brief. These problems are known as the *ZDT1*, *ZDT2* and *ZDT3* (see Zitzler *et al.* (2000) for detailed descriptions) and correspond to convex, concave and discontinuous Pareto fronts, respectively. Each of them involved 30 optimization variables and was solved using *EA*, *MAEA* and the proposed *MAMA*. To cope with *MOO* problems, the *SPEA*2 fitness assignment algorithm was used in all of them. Each run was repeated 25 times with different random number generation (RNG) seeds. The statistical processing of results is based on the hypervolume performance indicator I_H, Huang *et al.* (2007). Using I_H, the front quality is quantified by the percentage of the volume (area) of a bounding box which is not dominated by the front. By definition, small I_H values denote high quality fronts.

For all optimization runs, the parent and offspring populations used in *EA*, *MAMA* and *MAEA* had $\mu = 60$ and $\lambda = 100$ members, respectively. For the *MAEA* and *MAMA*, the *IPE* phase (i.e. the use of metamodels) started when the database $\mathcal{D}$ (of previously evaluated individuals) size exceeded 400 ($\lambda_{\mathcal{D}} \geq 400$). In *MAMA*, the *LS* started simultaneously with the *IPE* phase. Also, at each generation $\lambda_e = 15$ and $\lambda_{LS}^{max} = 12$.

Tables 1, 2 and 3 present statistical measures of the performances of the three algorithms during the evolution and demonstrate the superiority of *MAMA*, compared to *MAEA* and *EA*. Statistical t-tests on the average I_H values prove that, with the same CPU cost, *MAMA* outperforms *EA* and *MAEA* for a confidence level of 99.9% and for all the tested problems. Table 4 shows the outcomes of the t-test analysis of the *ZDT3* results. The reader may also refer to figures 5, 6 and 7 for a visual comparison of the optimal fronts obtained at the same CPU cost. Note that, in all cases, more than 60% of the individuals selected for *LS* were improved at least with respect to one of the objectives.

3.2. *Design of a Combined Cycle Power Plant*

This application is concerned with the optimal design of a combined cycle power plant (*CCPP*, Bonataki *et al.* (2004)). The power plant consists of a gas turbine (*GT*) producing a power output of 120 MW, a dual-pressure heat recovery steam generator (*HRSG*), a two-admission steam turbine (*ST*), feedwater tank, condenser, condensate and feedwater pumps and two generators (G_1 and G_2), as in figure 8. At the exit of the *GT*, the flue gas is cooled by traversing seven high- and low-pressure heat exchangers in the *HRSG*. Superheated high- and low-pressure steam is generated which is, then, expanded in the *ST*.

The 13 design parameters in total are classified to those determining the *GT* performance (pressure ratio, air-to-fuel ratio, polytropic efficiencies of the compressor and turbine, etc.) and those related to the *HRSG* design variables (heat exchange areas,

For Peer Review Only

operating pressures and temperatures for the working fluids).

In the present design, the two objectives to be maximized were the power plant efficiency and the produced power. The design was subject to a number of constraints related among other to the requirement that flue gas temperatures should exceed the water or steam temperature (by a safety margin) at different points along the *HRSG*. Due to the high number of constraints, many heavily penalized individuals entered $\mathcal{D}$ and this might cause troubles if the *RBF* networks were trained on penalized responses. For example, to approximate the score of a feasible solution which, in the design space, is located close to even a single infeasible solution, the presence of the latter in the *RBF* network training set may heavily mislead its prediction. The situation may become even worse, if the same *RBF* network was also used to drive *LS*. This problem is alleviated since the *RBF* network that supports *LS* is trained on the strength function values S instead of the penalized cost values. This is a noticeable advantage of the proposed *MAMA*.

As in the previous cases, the proposed *MAMA* is compared with *MAEA*s and *EA*s, based on three individual runs (using different RNG seeds). The core (μ,λ)-EA used in *MAMA* and *MAEA* had a common configuration: $\mu=40$, $\lambda=60$; *LS* and the *IPE* phase initiated when $\lambda_{\mathcal{D}} \geq 400$. Also, $\lambda_e=15$, $\lambda_{LS}^{max}=10$.

Average I_H indicator values and the range of deviation around the mean for a 66% probability are shown in figure 9. A t-test analysis on the final results (for 10000 exact evaluations) proved that, for 99.9% probability and irrespective of the RNG seed, *MAMA* reaches better optimal fronts than *EA* or *MAEA* (*MAMA* vs. *EA*: $t_0=13.03$ and *MAMA* vs. *MAEA*: $t_0=3.42$).

Figure 10 presents the fronts of non-dominated solutions obtained using the three algorithms at the cost of 500, 3000, 7000 and 10000 exact evaluations, for one of the three RNG seeds. It is obvious that *MAMA* performs better than the other two algorithms. In figure 11, some statistics on the performance of the *LS* process are shown; during the evolution, an average percentage of 45% of the members in $\mathcal{P}_{LS}$ were improved by *LS*, for at least one of the objectives.

3.3. *Two Operating Point Optimization of a Compressor Cascade Airfoil*

The second engineering test case is concerned with the design of an optimal compressor cascade airfoil at two operating points, with two objectives. The first objective is the minimization of the total pressure loss coefficient ω at the design point OP_1 (inlet flow angle $\alpha_1^{(OP_1)}=47^o$, inlet Mach number $M_1^{(OP_1)}=0.618$ and Reynolds $R_e^{(OP_1)}=8.41\cdot 10^5$). The second objective is the minimization of the same quantities at a different operation point OP_2 ($\alpha_1^{(OP_2)}=52^o$, $M_1^{(OP_2)}=0.621$ and $R_e^{(OP_2)}=7.63\cdot 10^5$). The cascade streamtube thickness before the leading edge (*LE*) and after the trailing edge (*TE*) was fixed to 1.0 and ∼0.9, respectively, Axial-Velocity-Density-Ratio with a linear variation in between *LE* and *TE* (*AVDR* =∼0.9).

The objective function to be minimized was expressed by $\omega=\frac{p_{t2,is}-p_{t2}}{p_{t1}-p_1}$, where p and p_t stand for static and total pressure, the subscripts 1 and 2 correspond to the cascade inlet and exit and *is* denotes isentropic flow. Drela's *MISES* analysis software for cascades, Drela and Giles (1987), was used for the evaluation of candidate airfoils during the optimization. *MISES* is based on an integral boundary layer method coupled with an external inviscid flow solver.

All candidate designs were subject to geometrical and flow constraints, related to

For Peer Review Only

the airfoil thickness (not allowed to exceed some minimum values at several chordwise positions) and the flow turning (which should exceed $\Delta\alpha_{12}^{(OP_1)}=24^o$ and $\Delta\alpha_{12}^{(OP_2)}=27^o$).

The airfoil pressure and suction sides were parameterized using two Bezier polynomial curves, with 7 control points each, giving rise to 20 design variables in total. On both curves, the first and last control points were kept fixed whereas the remaining five control points were free to vary within user-defined bounds.

The optimization was carried out by implementing a core (μ,λ)-EA (20,40) for all algorithms (*EA*, *MAEA* and *MAMA*); *LS* and *IPE* phase started at $\lambda_{\mathcal{D}}=300$, with $\lambda_e=8$ and $\lambda_{LS}^{max}=4$.

Figure 12 illustrates the non-dominated front computed by the three algorithms at the same cost of 5000 calls to the *MISES* software and the "convergence" history expressed in terms of the hypervolume indicator I_H. Note that, during the first 1000 evaluations, the two metamodel-assisted algorithms showed an equally efficient performance, compared to *EA*s. After 1500 evaluations, *MAMA* descended further though *MAEA*s and *EA*s drifted slowly without noticeable improvement in I_H. The elite front resulted from the *MAMA* after 5000 evaluations outperforms all the rest. For the optimal front computed by *MAMA* shown in figure 12 (right), its edge points were post-processed and are shown in figure 13 (contour profiles, isentropic Mach number distribution along the airfoil).

4. Conclusions – Comments

This paper presents a Metamodel-Assisted Memetic Algorithm which is suitable for multi-objective, constrained optimization problems. The proposed *MAMA* is advantageous since, as shown in the examined cases, reduces the CPU cost compared to conventional *MA*s or *MAEA*s. It was beyond the scope of this paper to compare this population-based stochastic algorithm with any other deterministic search algorithm. Such a comparison would simply come up with the well-known differences in efficiency and effectiveness of the two "rival" classes of methods. The purpose of this paper was to demonstrate the *MAMA*s superiority with respect to two methods (*EA*s and *MAEA*s) of the same class.

The CPU reduction was achieved by implementing local metamodels, trained on the fly on a small number of neighboring individuals, separately for each new candidate solution. The metamodels support both global and local search. During the local search, a novelty of our method is that the metamodel training is based on strength values (scalars that represent the number of offspring dominated by an individual) rather than objective function values. In constrained problems, where the objective function might be penalized due to the violation of constraints and/or problems where the objective function has a rugged landscape, this is important since penalties or abrupt landscape slopes do not affect the metamodel predictions.

We have chosen to use *RBF* networks which often produce better fits to various types of functions, compared to other rival techniques (such as polynomial regression or the kriging metamodel; see Jin and Chen (2001)); however, any other metamodel can be used instead. Depending on the problem, gradients computed using the *RBF* networks may be inaccurate but this doesn't affect the good performance of our method. This is attributed to two key features of our method (a) each *RBF* network corresponds to a small part of the search space, where the landscape is not usually expected to be rugged enough and (b) the landscape modeled by networks is the strength function. Therefore, to the authors' experience, inaccurate predictions of the strength gradient from the *RBF* network, during the local refinements, is not risky. According to the results shown here

For Peer Review Only

as well as other applications studied using this method, this neither imposes limitations to the applicability of the proposed method nor increases its CPU cost (due to the fact that during each refinement, either successful or failed due to *RBF* network inaccuracies, the extra CPU cost is that of a single evaluation).

5. Acknowledgment

This work was co–funded by the PENED03 program (Measure 8.3 of the Operational Program Competitiveness, of which 80 % is European Commission and 20 % national funding), under project number 03ED111.

References

Bonataki, E., Georgoulis, L., and Giannakoglou, C.G.K.C., 2004. Optimal design of combined cycle power plants based on gas turbine performance data. *In*: *ERCOFTAC Design Optimization: Methods & Applications*, Athens, Greece, 31 March - 2 April 2004., Athens.

Bosman, A. and de Jong, D.E., 2005. Exploiting gradient information in numerical multi–objective evolutionary optimization. *In*: *GECCO '05: Genetic and Evolutionary Computation Conference*, Washington DC, USA New York, NY, USA: ACM, 755–762.

Buche, D., Schraudolph, N.N., and Koumoutsakos, P., 2005. Accelerating evolutionary algorithms with gaussian process fitness function models. *IEEE Transactions on Systems, Man and Cybernetics – Part C: Applications and Reviews*, 35, 183–194.

Dawkins, R., 1976. *The selfish gene*. Oxford: Oxford University Press.

Drela, M. and Giles, M., 1987. Viscous-inviscid analysis of transonic and low Reynolds number airfoils. *AIAA Journal*, 25 (10), 1347–1355.

Giannakoglou, K., Giotis, A., and Karakasis, M., 2001. Low-Cost Genetic Optimization based on Inexact Pre-Evaluations and the Sensitivity Analysis of design parameters. *Journal of Inverse Problems in Engineering*, 9 (4), 389–412.

Giannakoglou, K.C., 2002. Viscous-inviscid analysis of transonic and low Reynolds number airfoils. *International Review Journal Progress in Aerospace Sciences*, 38, 43–7 6.

Hacker, K., 2002. Efficient global optimization using hybrid genetic algorithms. *In*: *9th AIAA//ISSMO: Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, USA.

Hart, W., 1994. Adaptive Global Optimization with Local Search. Thesis (PhD). University of California, USA.

Haykin, S., 1999. *Neural networks: A comprehensive foundation*. Prentice Hall International, Inc.

Huang, V.L., *et al.*, Problem Definitions for Performance Assessment of Multi-objective Optimization Algorithms. , 2007. , Technical report, Nanyang Technological University, Singapore Special Session on Constrained Real-Parameter Optimization.

Ishibuchi, H. and Murata, T., 1996. Multiobjective genetic local search algorithm. *In*: *IEEE International Conference on Evolutionary Computation*, Nagoya, Japan, 20-22 May 1996., 119–124.

Ishibuchi, H., Yoshida, T., and Murata, T., 2003. Balance between genetic search and

For Peer Review Only

local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE Transactions on Evolutionary Computation*, 7 (2), 204–223.

Jaszkiewicz, A., 2002. Genetic local search for multi-objective combinatorial optimization. *European Journal of Operational Research*, 137 (22), 50–71.

Jin, R. and Chen, W., 2001. Comparative studies of metamodeling techniques under multiple modeling criteria. *Structural and Multidisciplinary Optimization*, 23, 1–13.

Jin, Y., Olhofer, M., and Sendhoff, B., 2000. On evolutionary optimization with approximate fitness functions. *In*: D.W. et al., ed. *GECCO '00: Genetic and Evolutionary Computation Conference* Morgan Kaufmann, 786–793.

Jin, Y., Olhofer, M., and Sendhoff, B., 2002. A Framework for Evolutionary Optimization With Approximate Fitness Functions. *IEEE Transactions on Evolutionary Computation*, 6 (5), 481–494.

Karakasis, M. and Giannakoglou, K., 2005. On the use of metamodel-assisted, multiobjective evolutionary algorithms. *Engineering Optimization*, 38 (8), 941–957.

Keane, A. and Nair, P., 2005. *Computati onal approaches for aerospace design: The pursuit of excellence*. West Sussex: John Wiley & Sons, Inc.

Knowles, J. and Corne, D., 2000. M-PAES: A memetic algorithm for multiobjective optimization. *In*: *2000 Congress on Evolutionary Computation – CEC '00* IEEE Press, 325–332.

Krasnogor, N., 2002. Studies on the theory and design space of memetic algorithms. Thesis (PhD). UK.

Krasnogor, N. and Gustafson, S., 2002. Toward truly "memetic" memetic algorithms: discussion and proof of concepts. *In*: *Advances in Nature-Inspired Computation: The PPSN VII Workshops. PEDAL (Parallel, Emergent and Distributed Architectures Lab). University of Reading. ISBN 0-9543481-0-9. icalp.tex; 9/12/2003; 16:52; p.21 22 Natalio Krasnogor, Steven Gustafson.*

Liang, K., Yao, X., and Newton, C., 1999. Combining landscape approximation and local search in global optimization. *In*: *CEC '99, Congress on Evolutionary Computation*, Washington, USA IEEE Press, 1514–1520.

Liang, K., Yao, X., and Newton, C., 2000. Evolutionary search of approximated n-dimensional landscapes. *International Journal of Knowledge-based Intelligent Engineering Systems*, 4, 172–183.

Lim, D., *et al.*, 2008. Generalizing surrogate-assisted evolutionary computation. *IEEE Transactions on Evolutionary Computation, in press.*

Madsen, J., Shyy, W., and Haftka, R., 2000. Response Surface Techniques for Diffuser Shape Optimization. *AIAA Journal*, 38, 1512–1518.

Moscato, P., 1999. *Memetic algorithms: A short introduction.* McGraw-Hill Company.

Nocedal, J. and Wright, S.J., 1999. *Numerical Optimization.* Springer-Verlag.

Ong, Y.S., *et al.*, 2006. Classification of adaptive memetic algorithms: a comparative study. *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, 36 (1), 141–152.

Ong, Y. and Keane, A., 2004. Meta-Lamarckian learning in memetic algorithms. *Evolutionary Computation, IEEE Transactions on*, 8 (2), 99–110.

Ong, Y., Nair, P., and Keane, A., 2000. Evolutionary Optimization of Computationally Expensive Problems via Surrogate Modeling. *AIAA Journal*, 41 (4), 687–696.

Shan, S. and Wang, G.G., 2005. An Efficient Pareto Set Identification Approach for Multiobjective Optimization on Black-Box Functions. *Journal of Mechanical Design*, 127 (5), 866–874.

Ulmer, H., Streichert, F., and Zell, A., 2003. Model-assisted steady-state evolution strate-

For Peer Review Only

gies. *In*: *GECCO '03: Genetic and Evolutionary Computation Conference*, Chicago, USA, Chicago, USA, 610–621.

Ulmer, H., Streichert, F., and Zell, A., 2004. Evolution strategies with controlled model assistance. *In*: *CEC '04, Congress on Evolutionary Computation*, Portland, USA, Portland, USA.

Zhou, Z., *et al.*, 2007a. Memetic algorithm using multi-surrogates for computationally expensive optimization problems. *Journal of Soft Computing*, 11 (10), 957–971.

Zhou, Z., *et al.*, 2007b. Combining global and local surrogate models to accelerate evolutionary optimization. *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, 37 (1), 66–76.

Zitzler, E., Deb, K., and Thiele, L., 2000. Comparison of multiobjective evolutionary algorithms: empirical results. *Evolutionary Computation*, 8, 173–195.

Zitzler, E., Laumans, M., and Thiele, L., SPEA2: Improving the Strength Pareto Evolutionary Algorithm for multiobjective optimization. , 2001. , Technical report, Zurich TIK–Report 103.

For Peer Review Only

6. Appendix

For the validation of the proposed method, three benchmark two-objective function minimization problems, namely the *ZDT1*, *ZDT2* and *ZDT3* (see Zitzler *et al.* (2000)) test problems, were solved. For all the *ZDT*-problems, the equations expressing the design space and the objective functions are given in equations 1 and 2. The functions $G(\vec{x})$ and $h(\vec{x})$ differ among the three test cases.

$$F_1(\vec{x}) = x_1 \quad \text{and} \quad F_2(\vec{x}) = h(\vec{x})G(\vec{x}) \tag{1}$$

$$\vec{x} \in \mathcal{R}^N, \ N = 30, \quad x_n \in [0,1], \ n = 1, N \tag{2}$$

The *ZDT1* problem has a convex optimal Pareto front, shown in the upper-left plot in figure 14. The functions $G(\vec{x})$ and $h(\vec{x})$ are given in equation 3.

$$G(\vec{x}) = 1 + \frac{9}{N-1}\sum_{n=2}^{N} x_n \ , \qquad h(\vec{x}) = 1 - \sqrt{\frac{F_1(\vec{x})}{G(\vec{x})}} \tag{3}$$

The *ZDT2* problem has a concave optimal Pareto front, shown in figure 14, upper-right. The functions $G(\vec{x})$ and $h(\vec{x})$ are given in equation 4

$$G(\vec{x}) = 1 + \frac{9}{N-1}\sum_{n=2}^{N} x_n \ , \qquad h(\vec{x}) = 1 - (\frac{F_1(\vec{x})}{G(\vec{x})})^2 \tag{4}$$

ZDT3 leads to a discontinuous Pareto front shown in the lower plot in figure 14, achieved by setting $x_n = 0, \ n = 2, N$. The functions $G(\vec{x})$ and $h(\vec{x})$ are given in equation 5.

$$G(\vec{x}) = 1 + \frac{9}{N-1}\sum_{n=2}^{N} x_n \ , \qquad H(\vec{x}) = 1 - \sqrt{\frac{F_1(\vec{x})}{G(\vec{x}}} - \frac{F_1(\vec{x})}{G(\vec{x})} sin\left(10\pi F_1\left(\vec{x}\right)\right) \tag{5}$$

For Peer Review Only

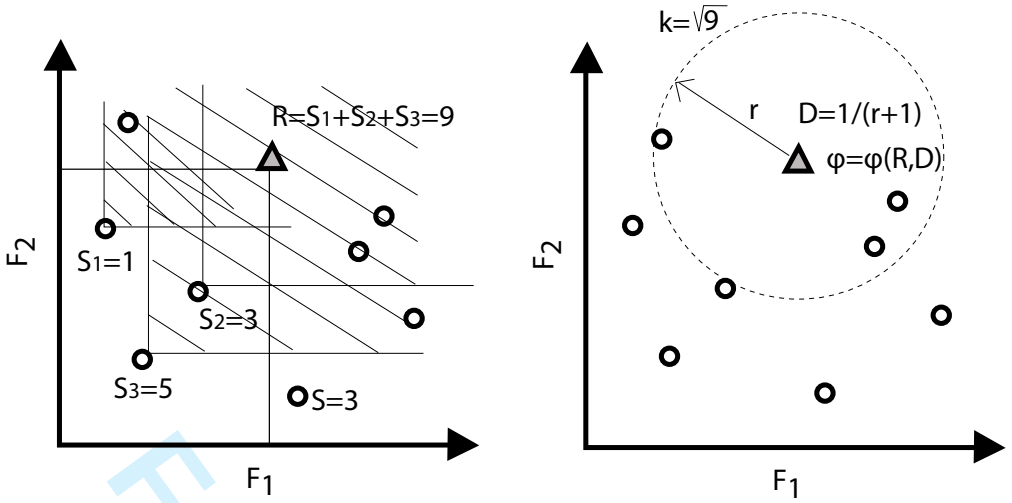


Figure 1. Calculation of the fitness ϕ for *MOO* cases, according to the raw fitness R, the strength S and the density D. The strength S of any individual equals the number of the rest members in $\mathcal{P}_\lambda$ it dominates. For the individual shown with a triangle, its R value is determined by the summation of strengths of its dominators, i.e. $R = S_1 + S_2 + S_3$. Its density D is adversely proportional to the distance from its closest $\sqrt{\lambda}$ neighbor in the objectives space. Finally, its fitness ϕ equals the summation of R plus D. For further details on the ϕ function, the reader is addressed to Zitzler *et al.* (2001).

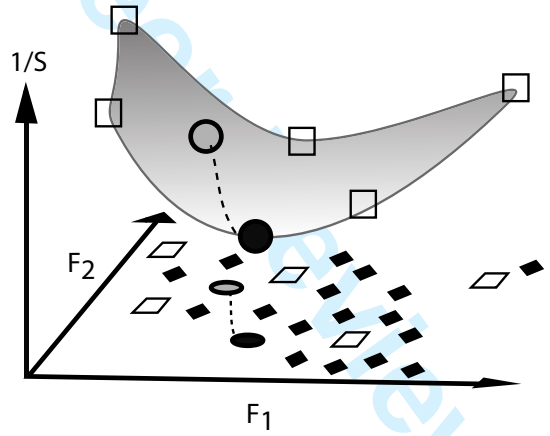


Figure 2. Example of the *LS* process. Training patterns are marked with empty squares. The individual to be refined is marked with the empty circle and the *LS* outcome with a black filled circle. Black diamonds correspond to the offspring population.

For Peer Review Only

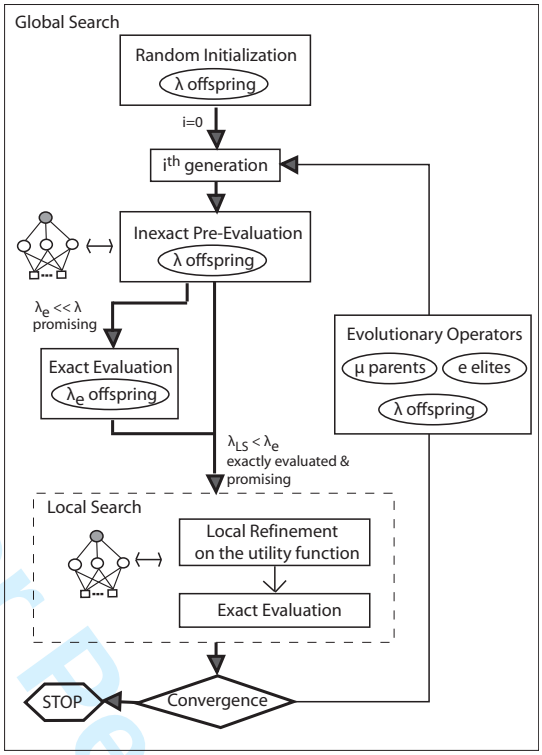


Figure 3. Flowchart of the proposed *MAMA* algorithm.

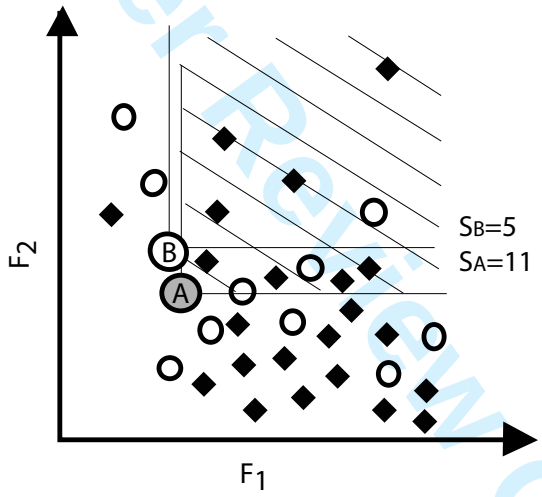


Figure 4. Example of the selection of individuals to undergo refinement. Individuals with circles represent individuals evaluated on the problem-specific tool ($\mathcal{P}_{e,g}$), whereas diamonds correspond to offpsring ($\mathcal{P}_{\lambda,g} \backslash \mathcal{P}_{e,g}$) evaluated only on the metamodel. Among A and B, A is the most favored to be selected for LS, since its strength is greater than the one of B and, thus, its refinement is more likely to lead to improvement.

For Peer Review Only

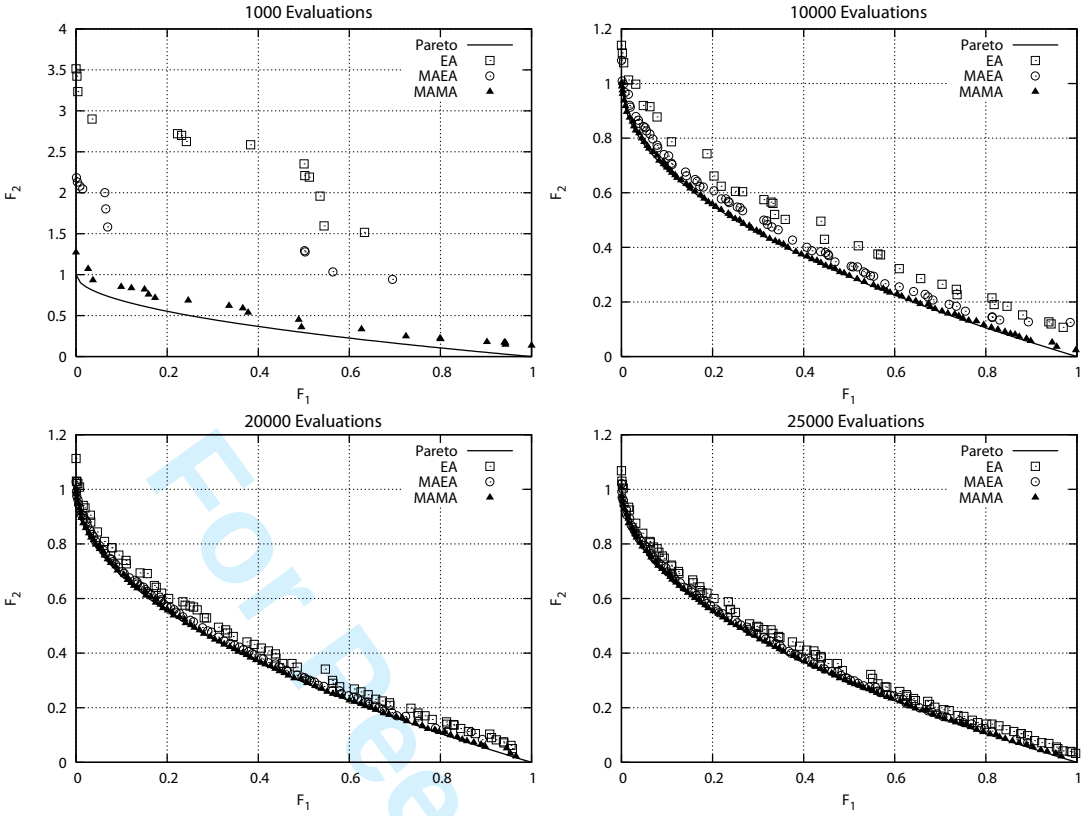


Figure 5. *ZDT1*: Fronts of the elite solutions for *EA*, *MAEA*, *MAMA* at the cost of 1000 (top-left), 10000 (top-right), 20000 (bottom-left) and 25000 (bottom-right) calls to the function evaluation tool for the best performing RNG seed. The exact Pareto front corresponds to the continuous line.

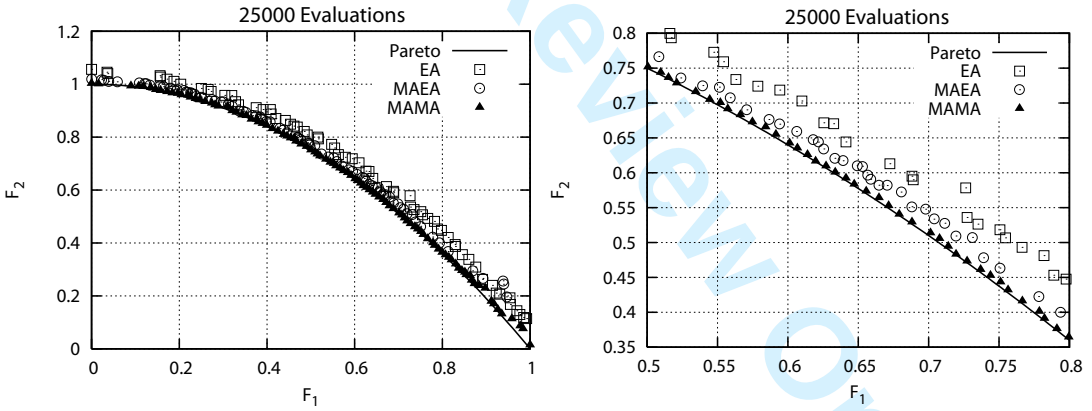


Figure 6. *ZDT2*: Fronts of the elite solutions for *EA*, *MAEA*, *MAMA* at the cost of 25000 (left) calls to the function evaluation tool for the best performing RNG seed. The right plot focuses on a part of the optimal front, to clearly show that the proposed *MAMA* approximates better the exact Pareto front (continuous line).

For Peer Review Only

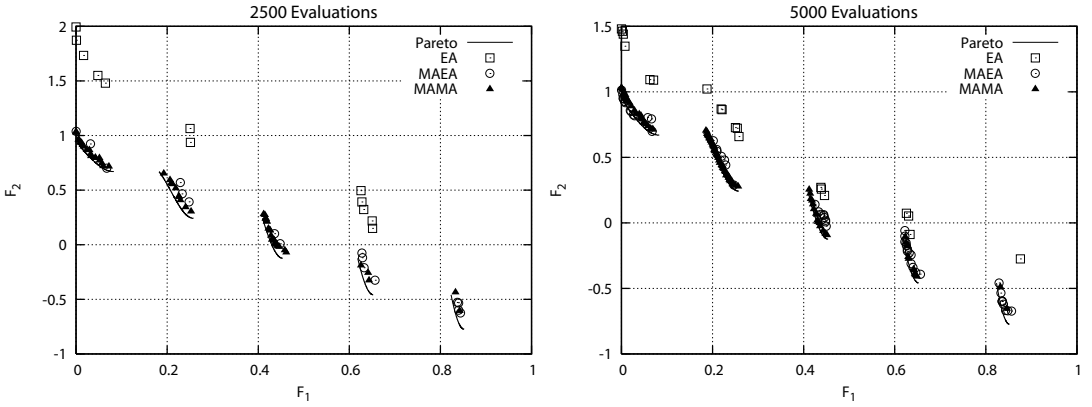


Figure 7. *ZDT3*: Fronts of the elite solutions for *EA*, *MAEA*, *MAMA* at the cost of 2500 (left) and 5000 (right) calls to the function evaluation tool for the best performing RNG seed. The exact Pareto front is also shown (continuous line).

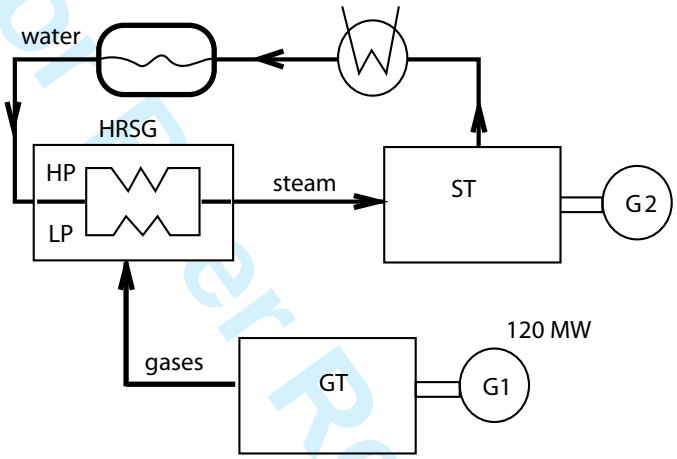


Figure 8. *Design of a CCPP:* Power plant configuration with a dual pressure heat recovery steam generator.

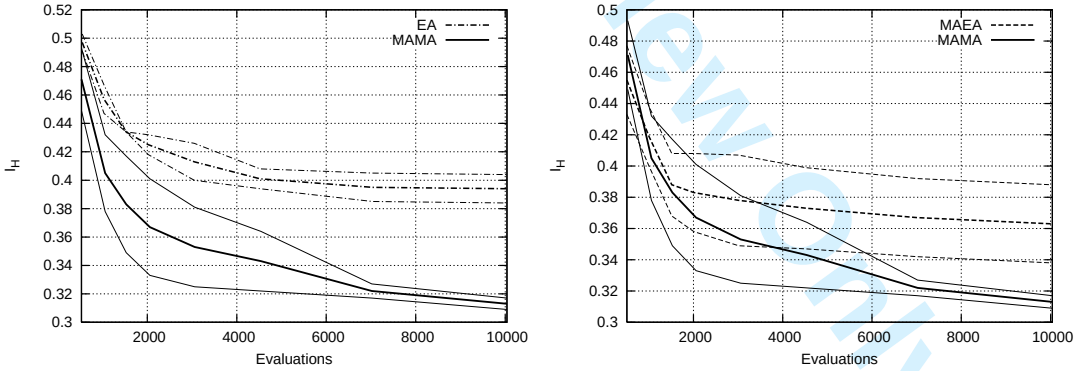


Figure 9. *Design of a CCPP:* Statistical measures for the performance of *EA* vs. *MAMA* (left) and *MAEA* vs. *MAMA* (right) for the thre three runs: average I_H values and the range of deviation (pairs of dashed lines) around the mean for a 66% probability are shown.

For Peer Review Only

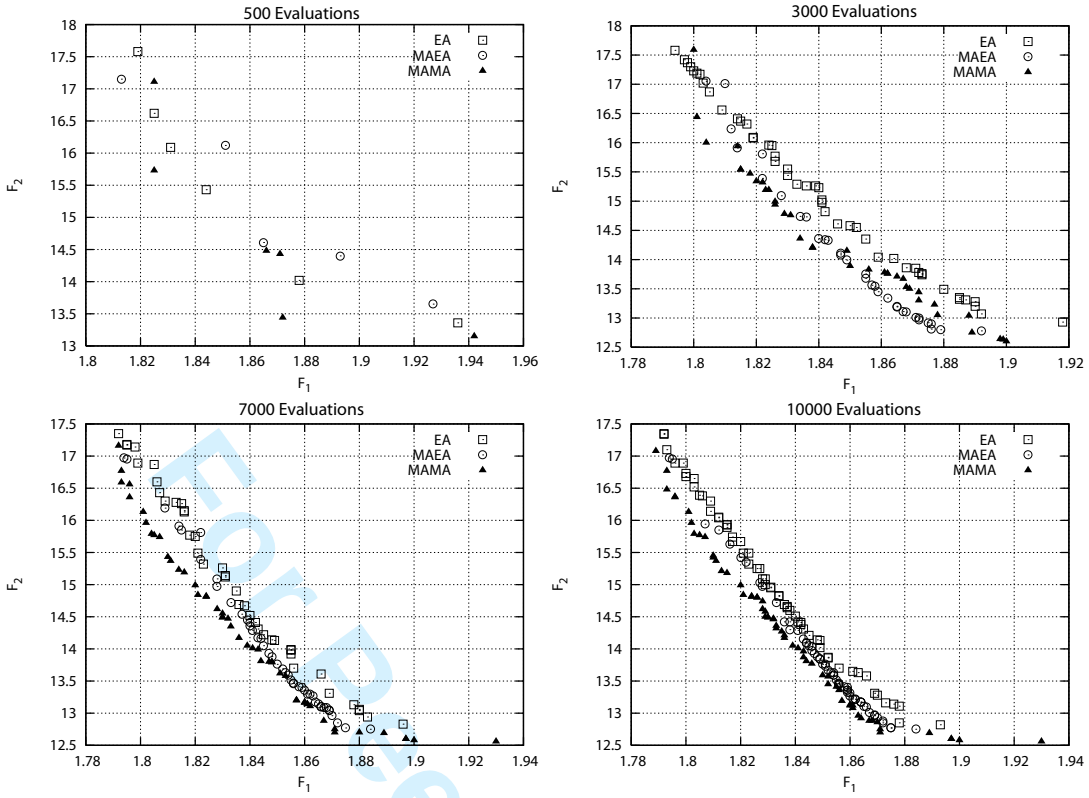


Figure 10. *Design of a CCPP:* Fronts of non-dominated solutions achieved using *EAs*, *MAEAs* and *MAMA* at 500, 3000, 7000 and 10000 exact evaluations.

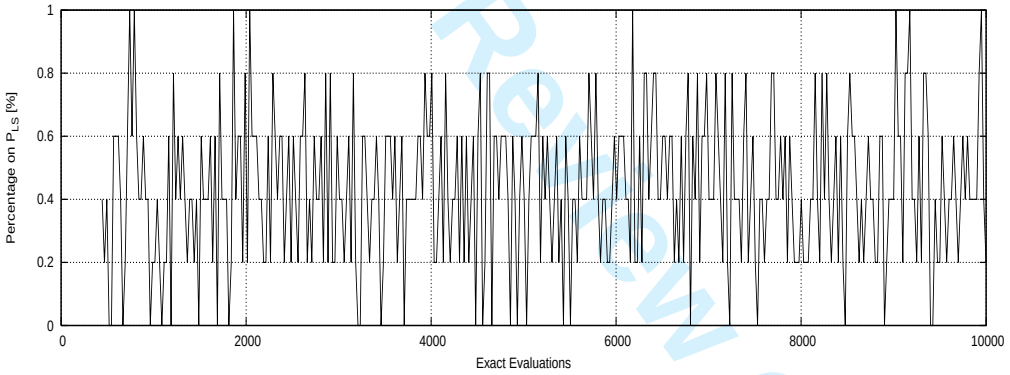


Figure 11. *Design of a CCPP: LS* statistics for the proposed *MAMA*: Percentage of totally or semi-improved (improved for all or at least one of the objectives, respectively) on the total number of individuals selected to undergo *LS*.

For Peer Review Only

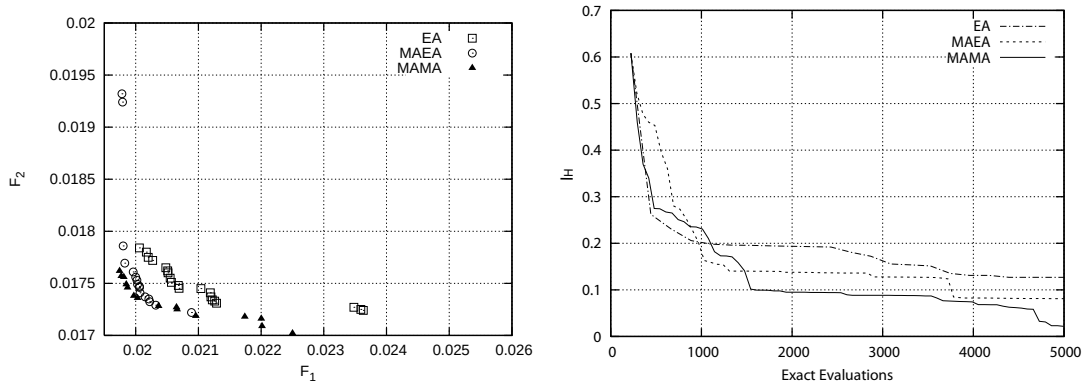


Figure 12. *Design of a compressor cascade at two operating points:* Left plot: Fronts of non-dominated solutions computed by the three methods at a cost of 5000 exact evaluations. Right plot: Convergence measured in terms of the dominated hypervolume I_H.

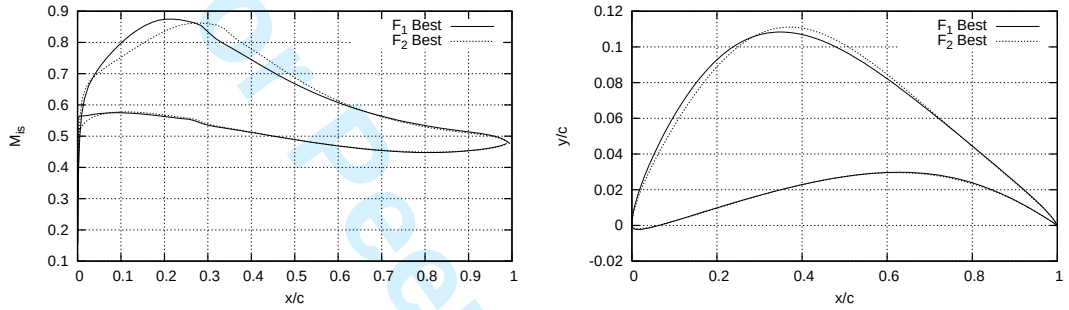


Figure 13. *Design of a compressor cascade at two operating points:* Comparison of the extreme Pareto individuals computed using *MAMA* at the cost of 5000 evaluations.

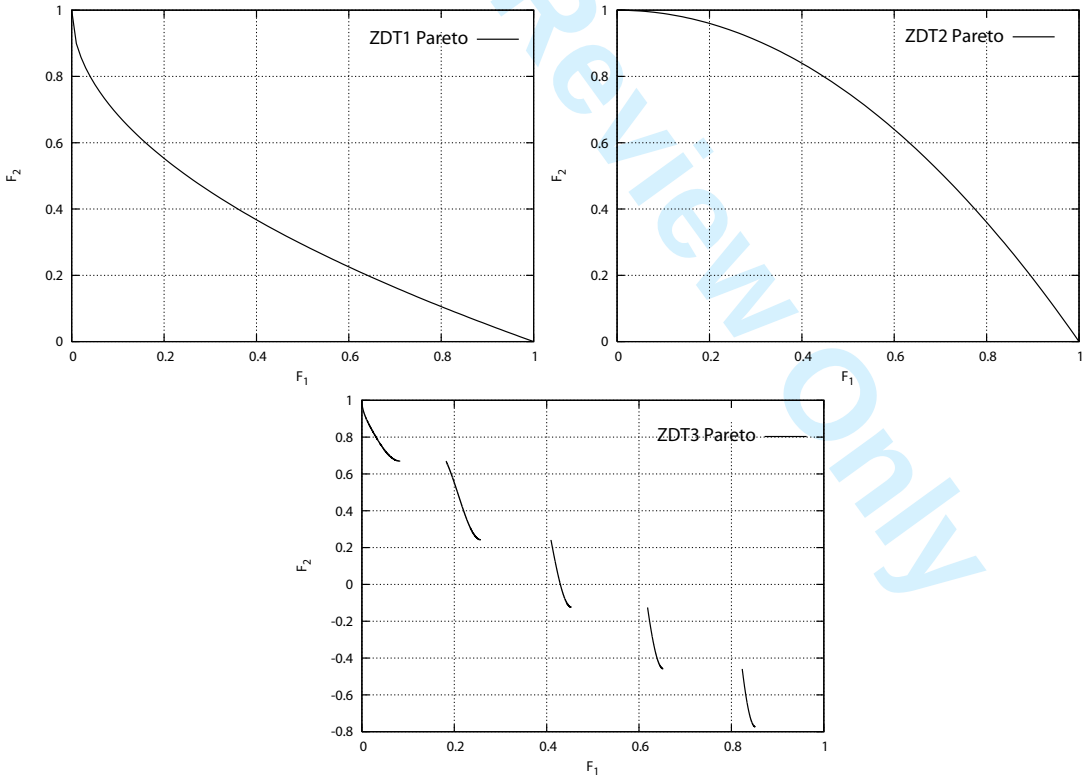


Figure 14. *Mathematical Benchmarks:* Analytical Pareto front of the *ZDT1* (up-left), the *ZDT2* (up-right) and the *ZDT3* (low) test problems, gained by setting $x_n = 0,\ n = 2, N$.

For Peer Review Only

Table 1. *ZDT1*: Statistical measures for the performance of *EA*, *MAEA* and the proposed *MAMA* for the 25 runs: average, best, median and worst I_H values along with the standard deviation for 5000, 15000 and 25000 evaluations on the problem-specific tool are shown. Statistical t-tests prove that the proposed *MAMA* outperforms *EA* and *MAEA*, at any "instant" of the evolution.

Algorithm	Evaluations	Average	Stdev	Best	Median	Worst
EA	5000	0.144	0.009	0.128	0.150	0.165
	15000	0.088	0.003	0.083	0.088	0.093
	25000	0.080	0.002	0.077	0.080	0.083
MAEA	5000	0.094	0.004	0.088	0.096	0.102
	15000	0.076	0.001	0.074	0.077	0.079
	25000	0.073	0.001	0.071	0.073	0.075
MAMA	5000	0.071	0.001	0.070	0.072	0.074
	15000	0.069	0.0004	0.069	0.070	0.071
	25000	0.069	0.0004	0.069	0.069	0.070

Table 2. *ZDT2*: Statistical measures for the performance of *EA*, *MAEA* and the proposed *MAMA* for 25 runs: average, best, median and worst I_H values for 5000, 15000 and 25000 evaluations on the problem-specific tool are shown.

Algorithm	Evaluations	Average	Stdev	Best	Median	Worst
EA	5000	0.240	0.009	0.223	0.239	0.255
	15000	0.163	0.004	0.156	0.163	0.169
	25000	0.151	0.002	0.147	0.150	0.154
MAEA	5000	0.192	0.018	0.166	0.186	0.216
	15000	0.155	0.014	0.144	0.175	0.201
	25000	0.149	0.011	0.141	0.164	0.184
MAMA	5000	0.164	0.035	0.138	0.200	0.262
	15000	0.142	0.012	0.136	0.155	0.186
	25000	0.140	0.009	0.136	0.146	0.173

For Peer Review Only

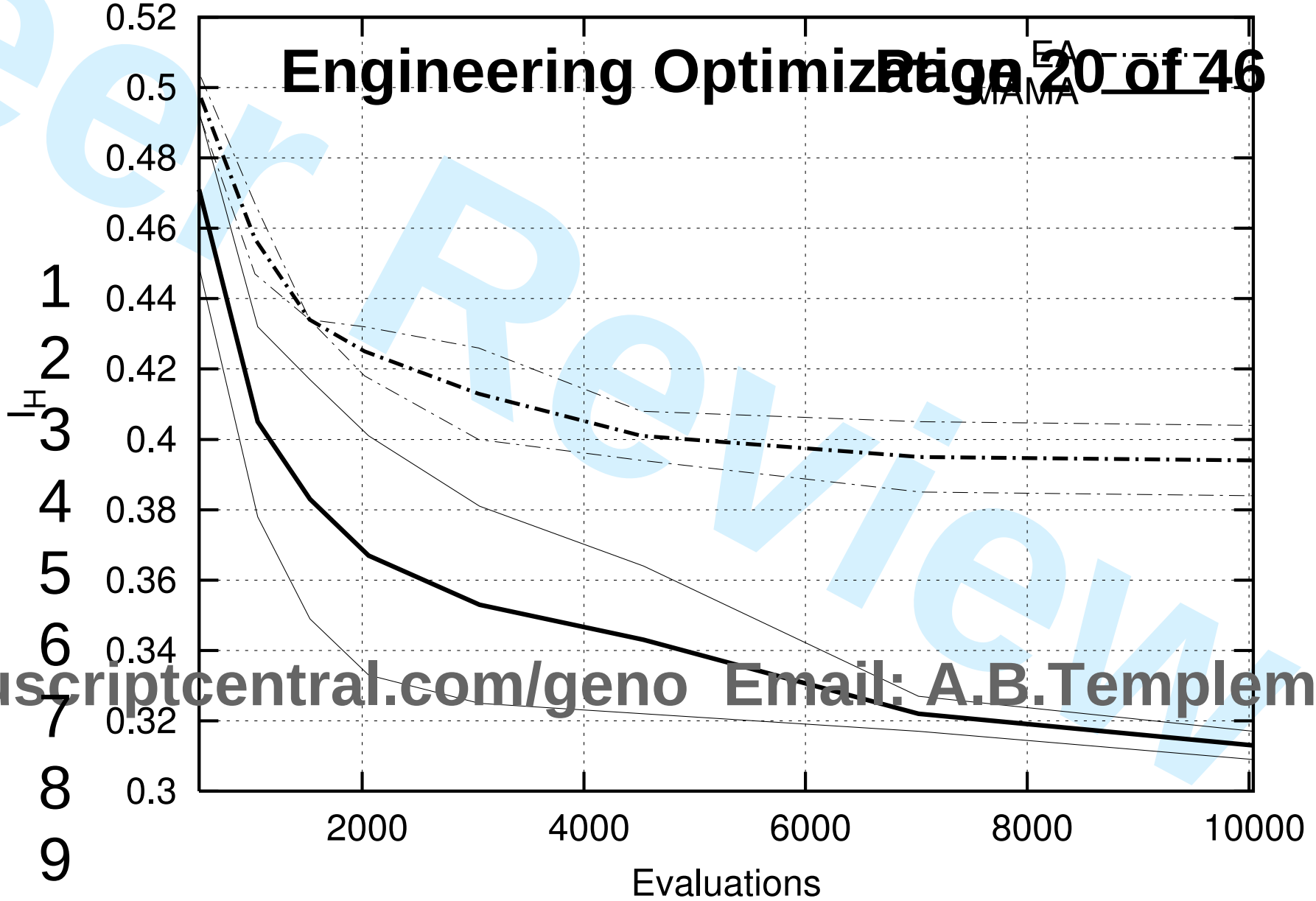
Table 3. *ZDT*3: Statistical measures for the performance of *EA*, *MAEA* and the proposed *MAMA* for 25 runs: average, best, median and worst I_H values for for 2500 and 5000 evaluations on the problem-specific tool are shown.

Algorithm	Evaluations	Average	Stdev	Best	Worst
EA	2500	0.27	0.04	0.23	0.31
	5000	0.19	0.02	0.17	0.21
MAEA	2500	0.16	0.03	0.13	0.22
	5000	0.14	0.02	0.12	0.22
MAMA	2500	0.13	0.02	0.12	0.16
	5000	0.12	0.01	0.11	0.13

Table 4. *ZDT3*: Results of the t-test analysis (values of the t_0 variable) for the three methods at several "instants" during the evolution, which prove that *MAMA* outperform *EA* and *MAEA* for a 99.9% probability.

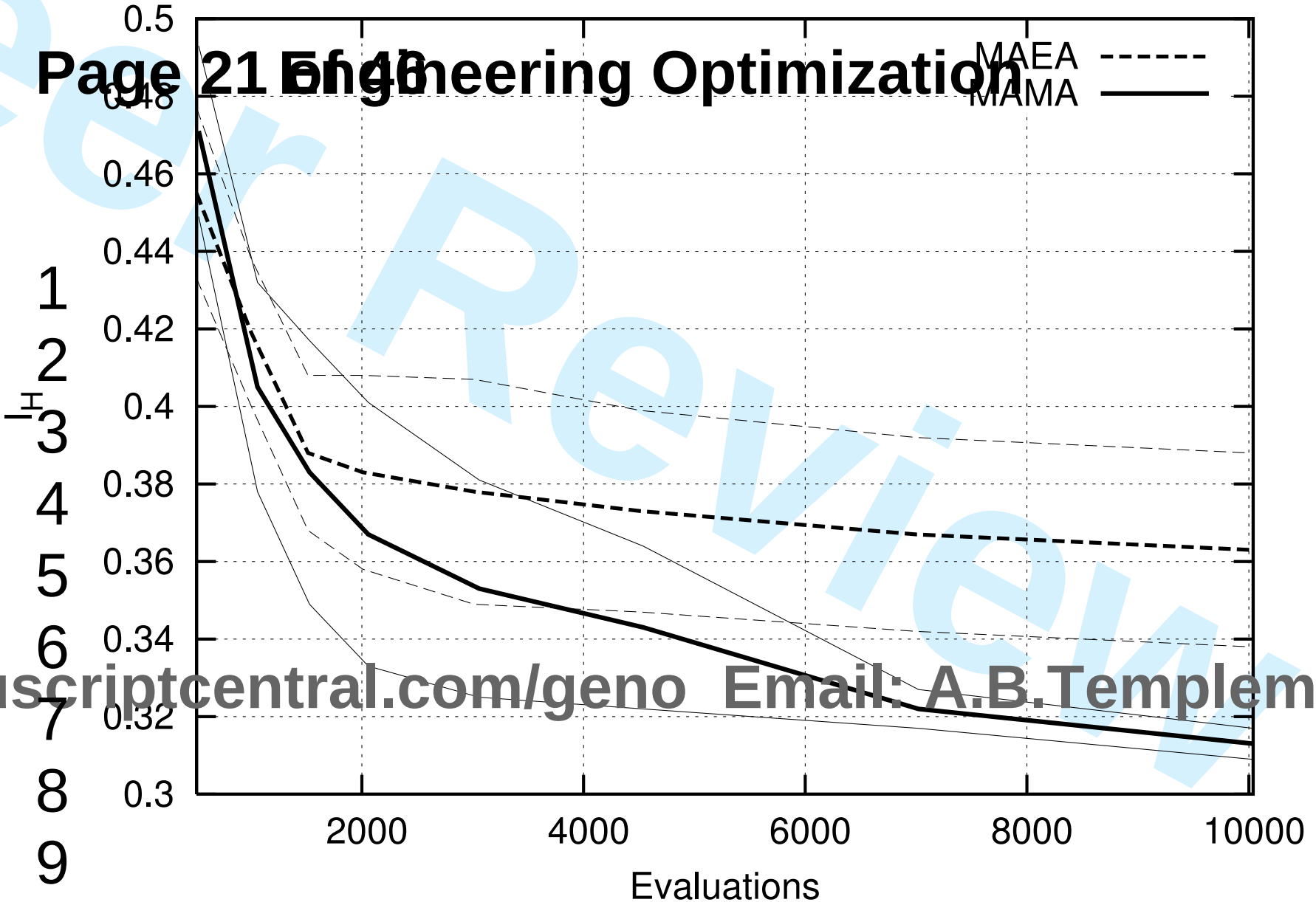
	500	1500	2500	5000
MAMA vs. EA	9.2	12.4	13.0	8.5
MAMA vs. MAEA	5.5	4.2	4.1	4.5

For Peer Review Only

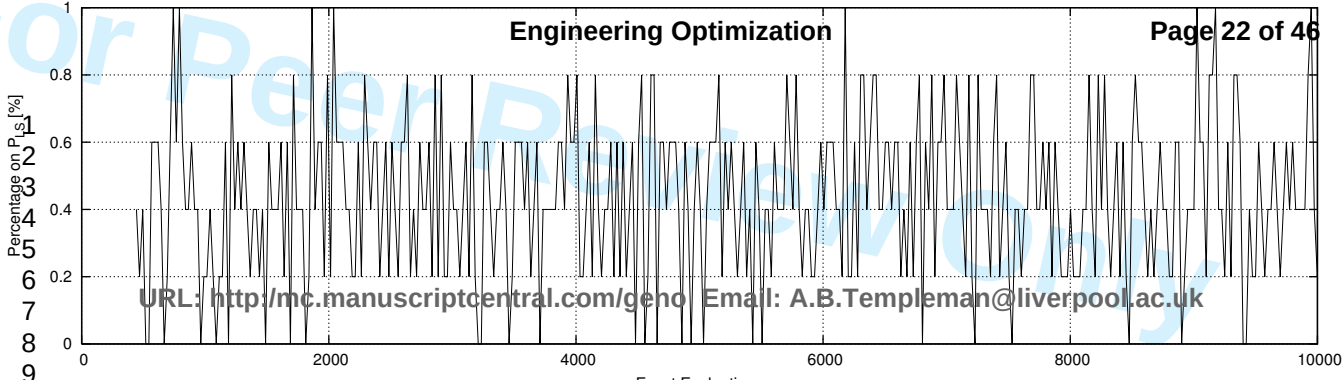


uscriptcentral.com/geno Email: A.B.Templem

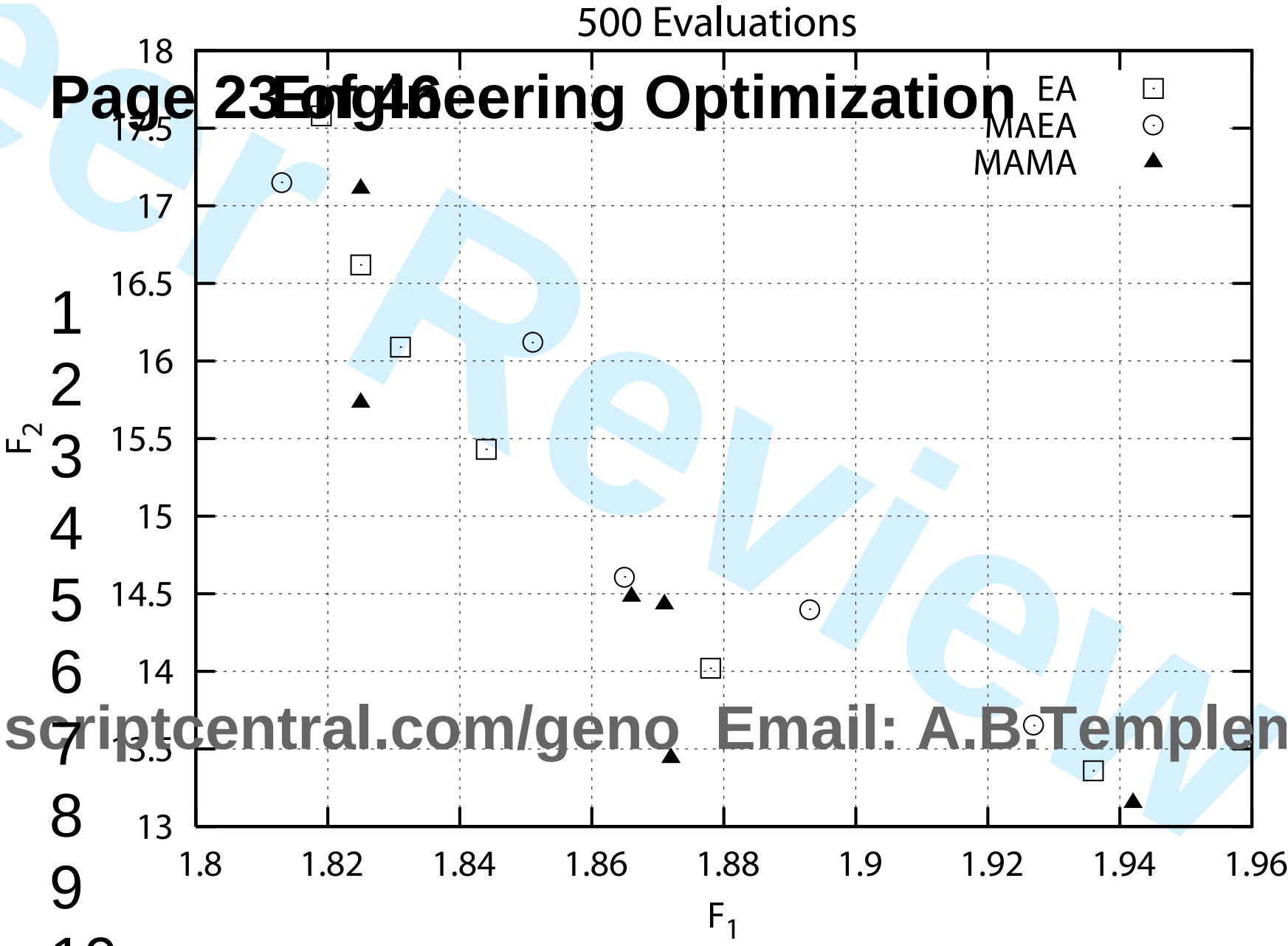
Peer Review



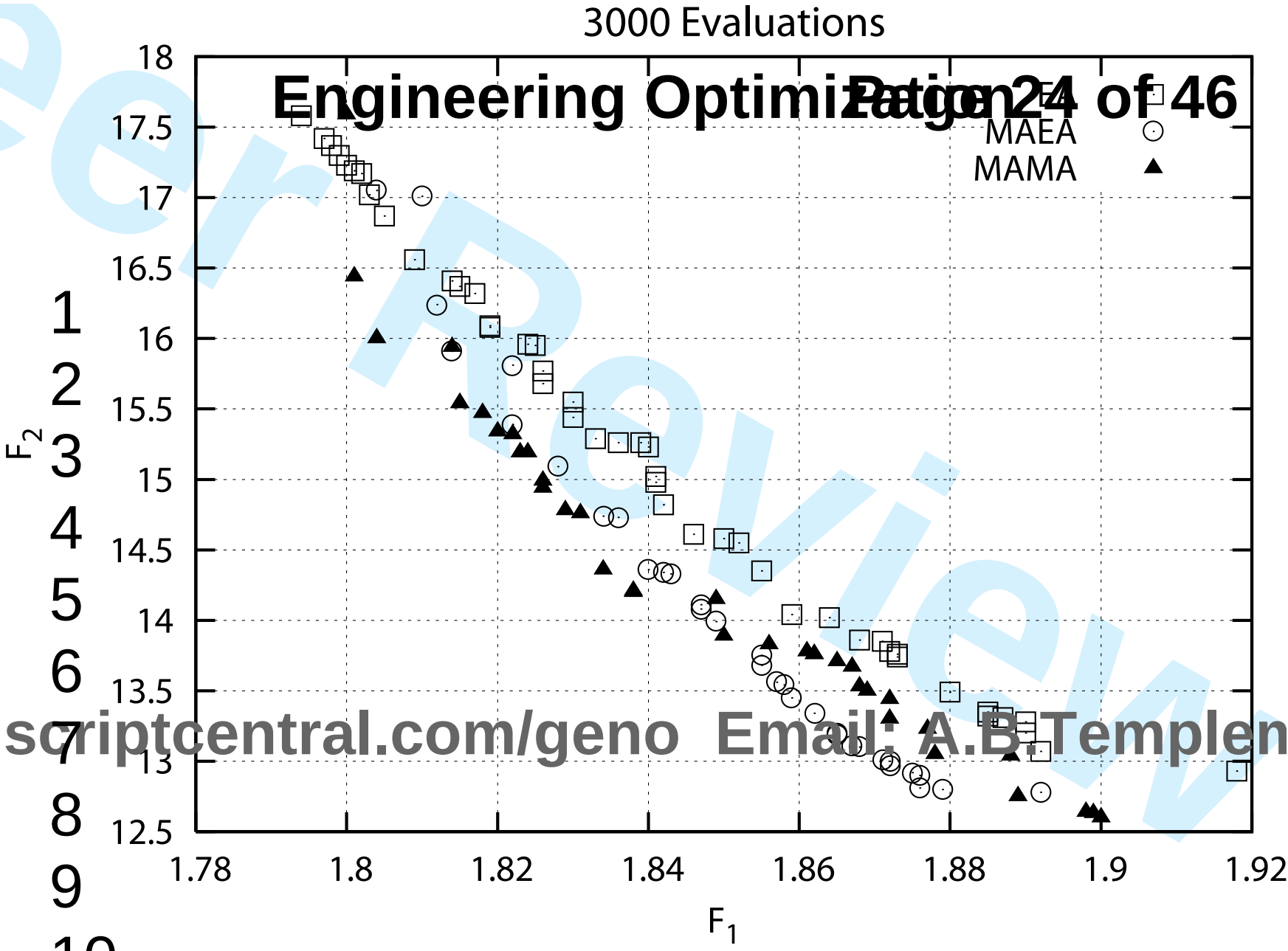
uscriptcentral.com/geno Email: A.B.Templem



For Peer Review Only
URL: http://mc.manuscriptcentral.com/geno Email: A.B.Templeman@liverpool.ac.uk

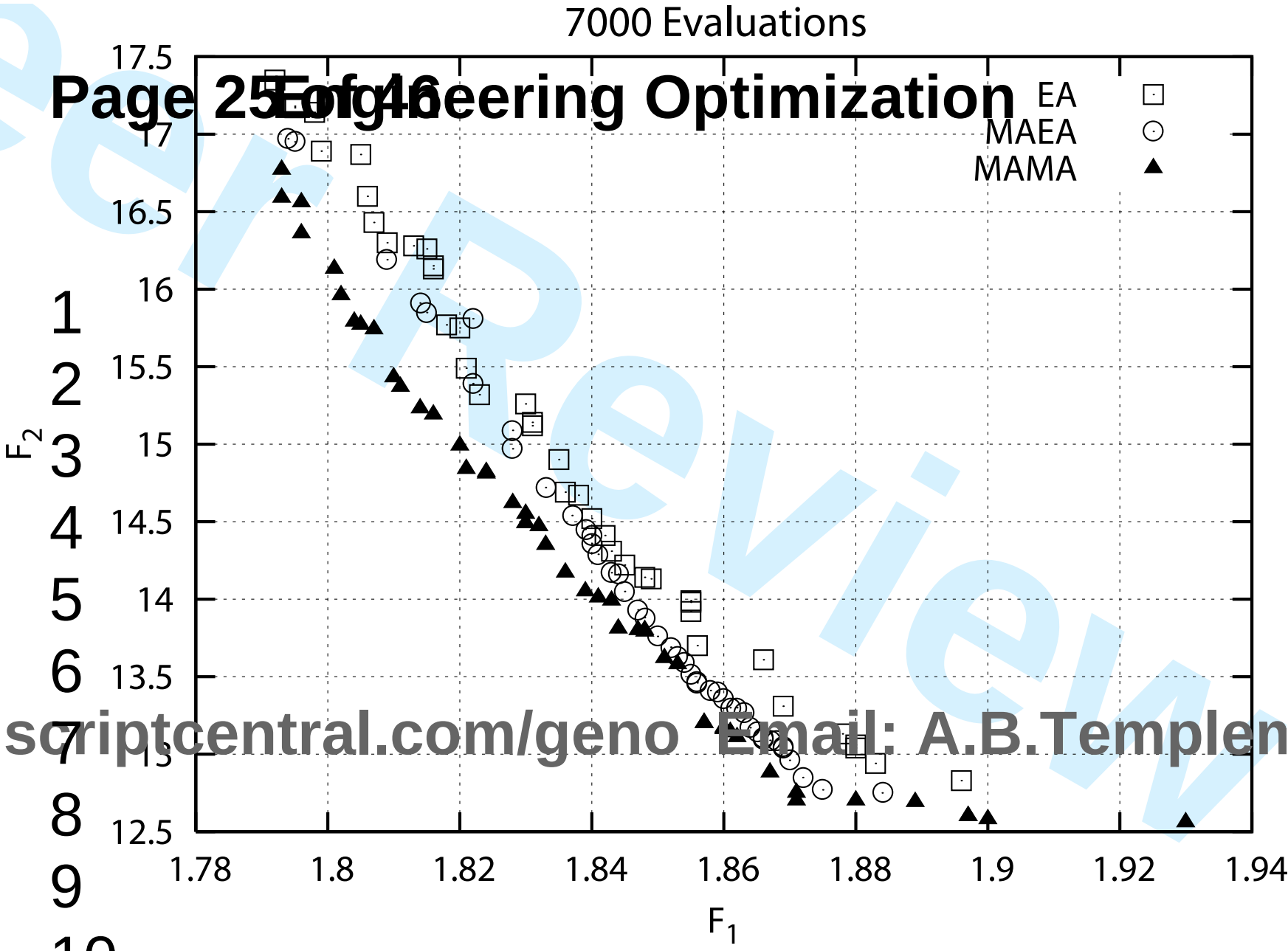


scriptcentral.com/geno Email: A.B.Templen

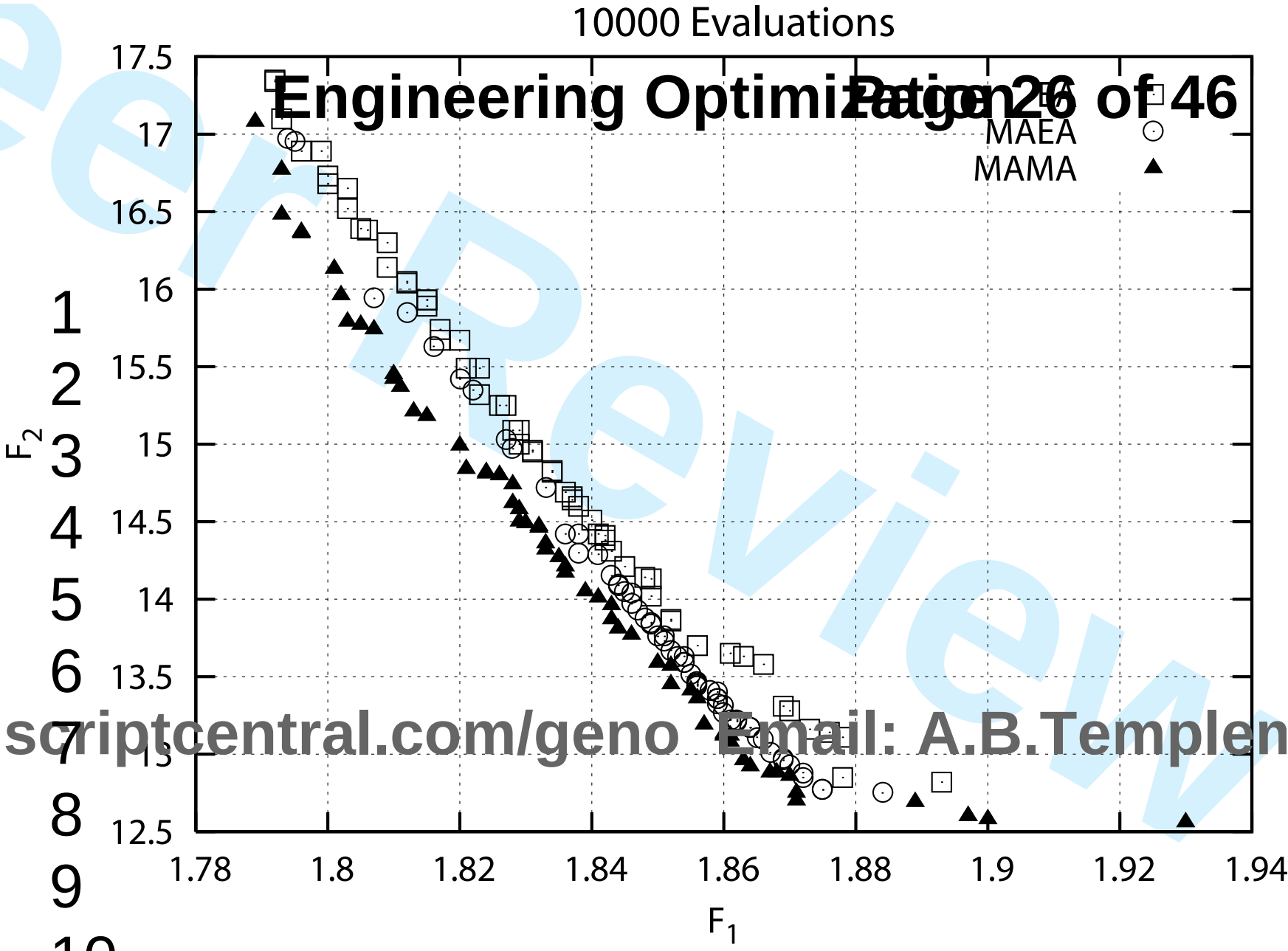


scriptcentral.com/geno Email: A.B.Templen

For Review

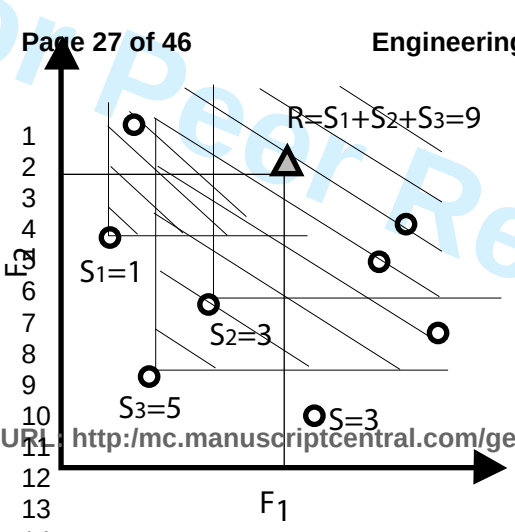


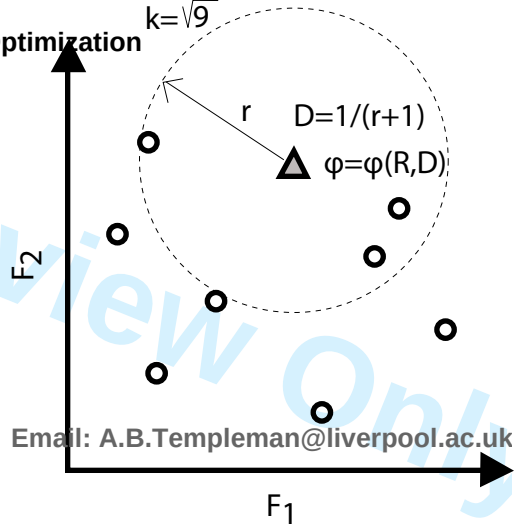
scriptcentral.com/geno Email: A.B.Templen



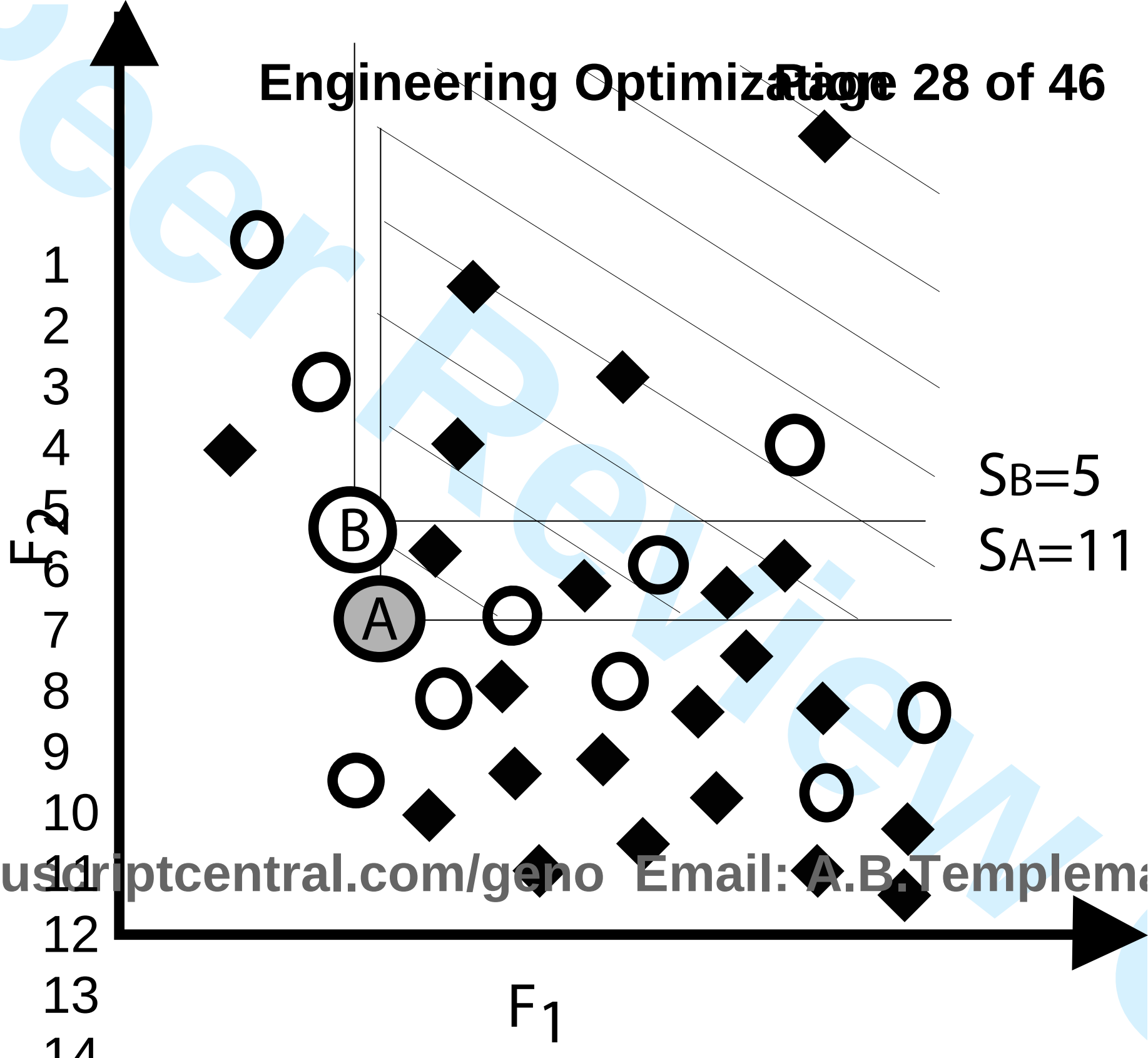
For Peer Review

scriptcentral.com/geno Email: A.B.Templen

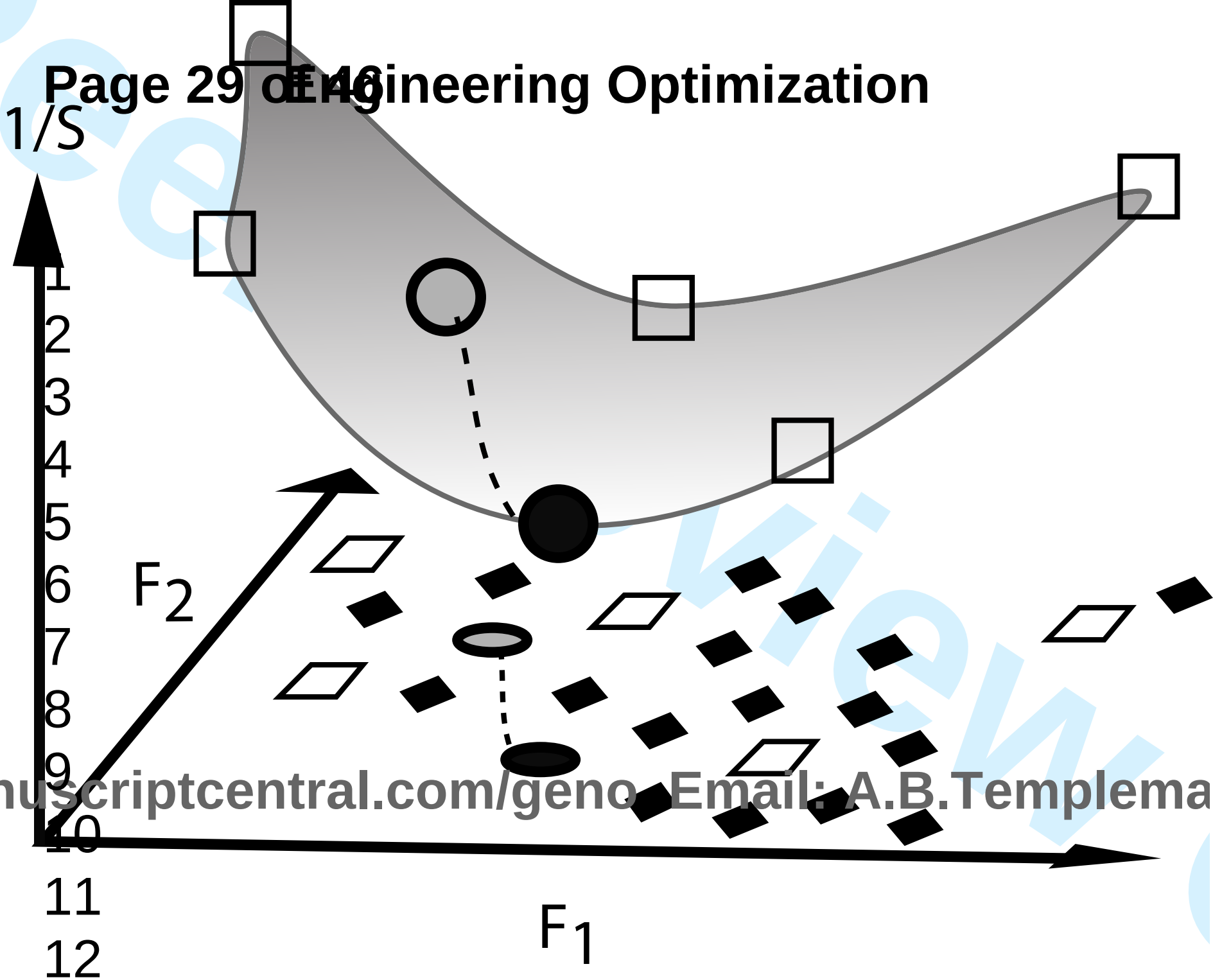




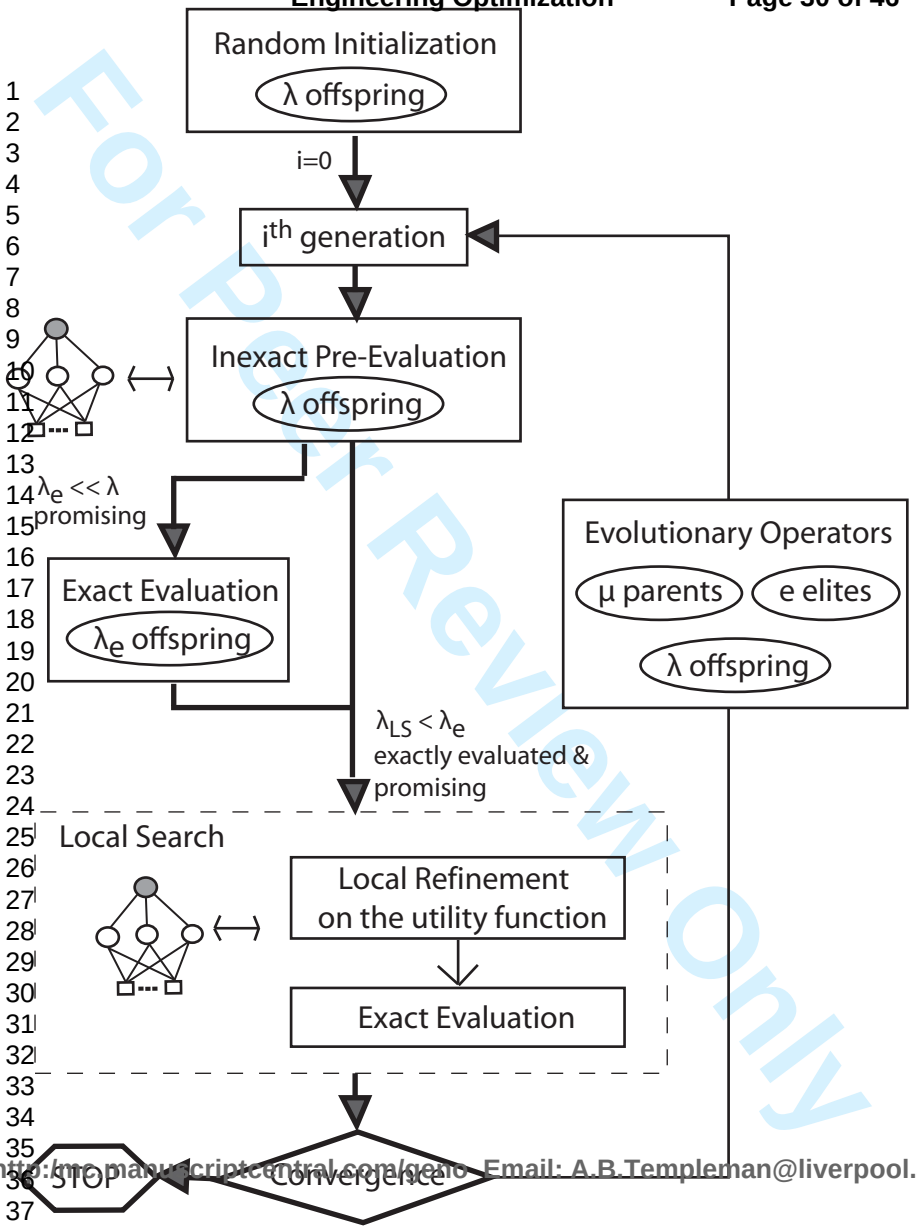
For Peer Review Only



Under Review
uscriptcentral.com/geno Email: A.B.Templema

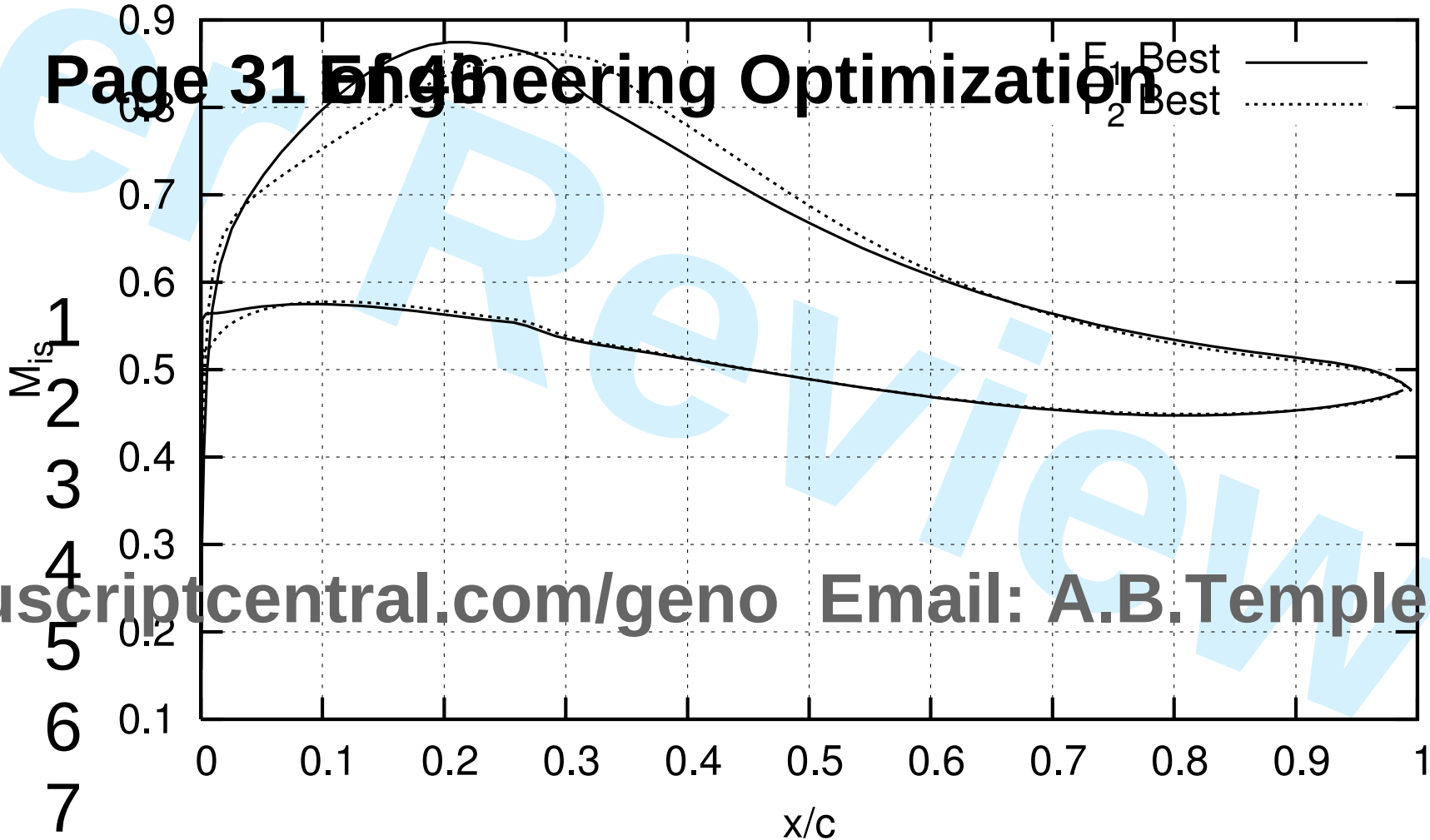


nuscriptcentral.com/geno Email: A.B.Templema

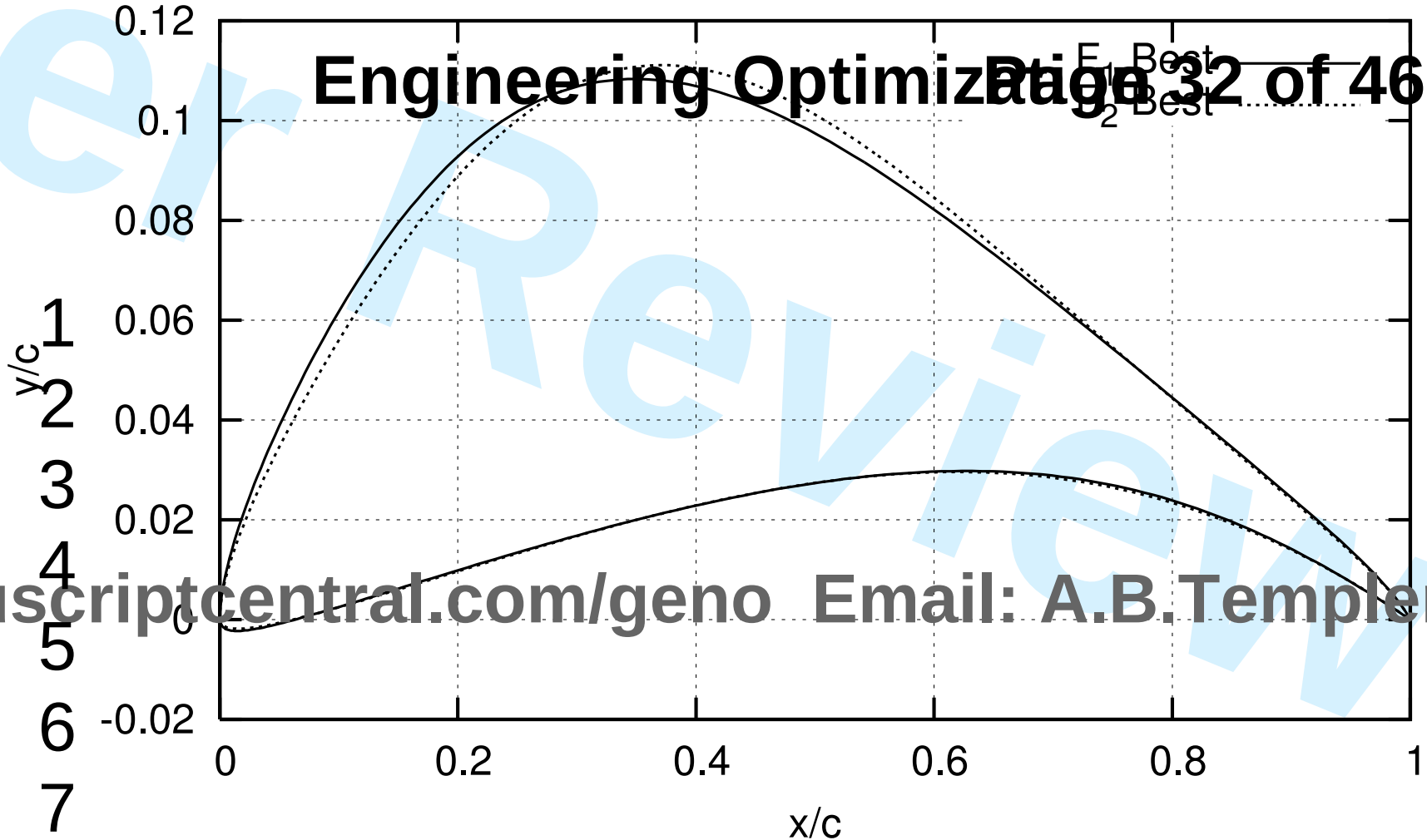


For Peer Review Only

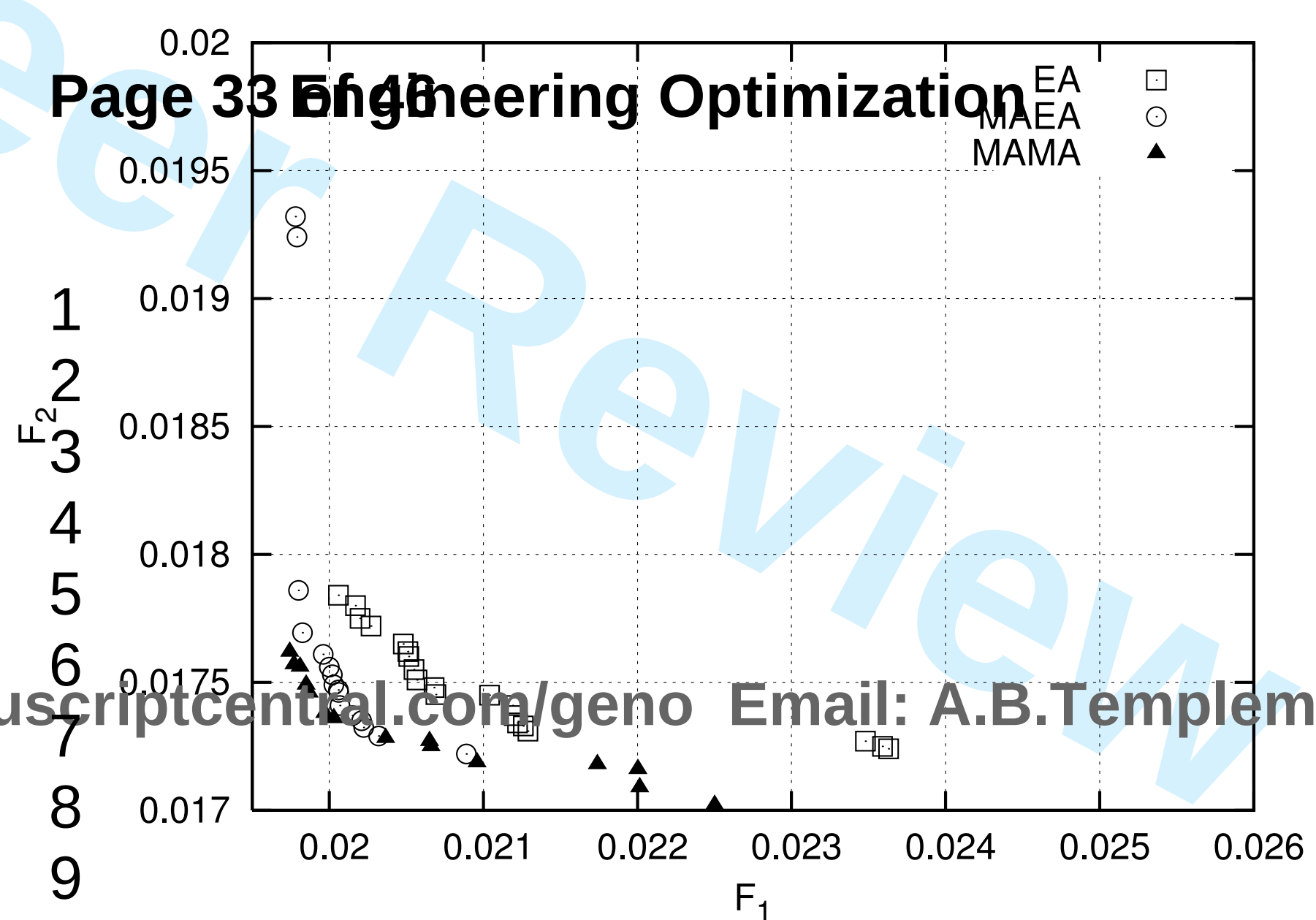
http://mc.manuscriptcentral.com/geno Email: A.B.Templeman@liverpool.a



For Review Only
uscriptcentral.com/geno Email: A.B.Templem

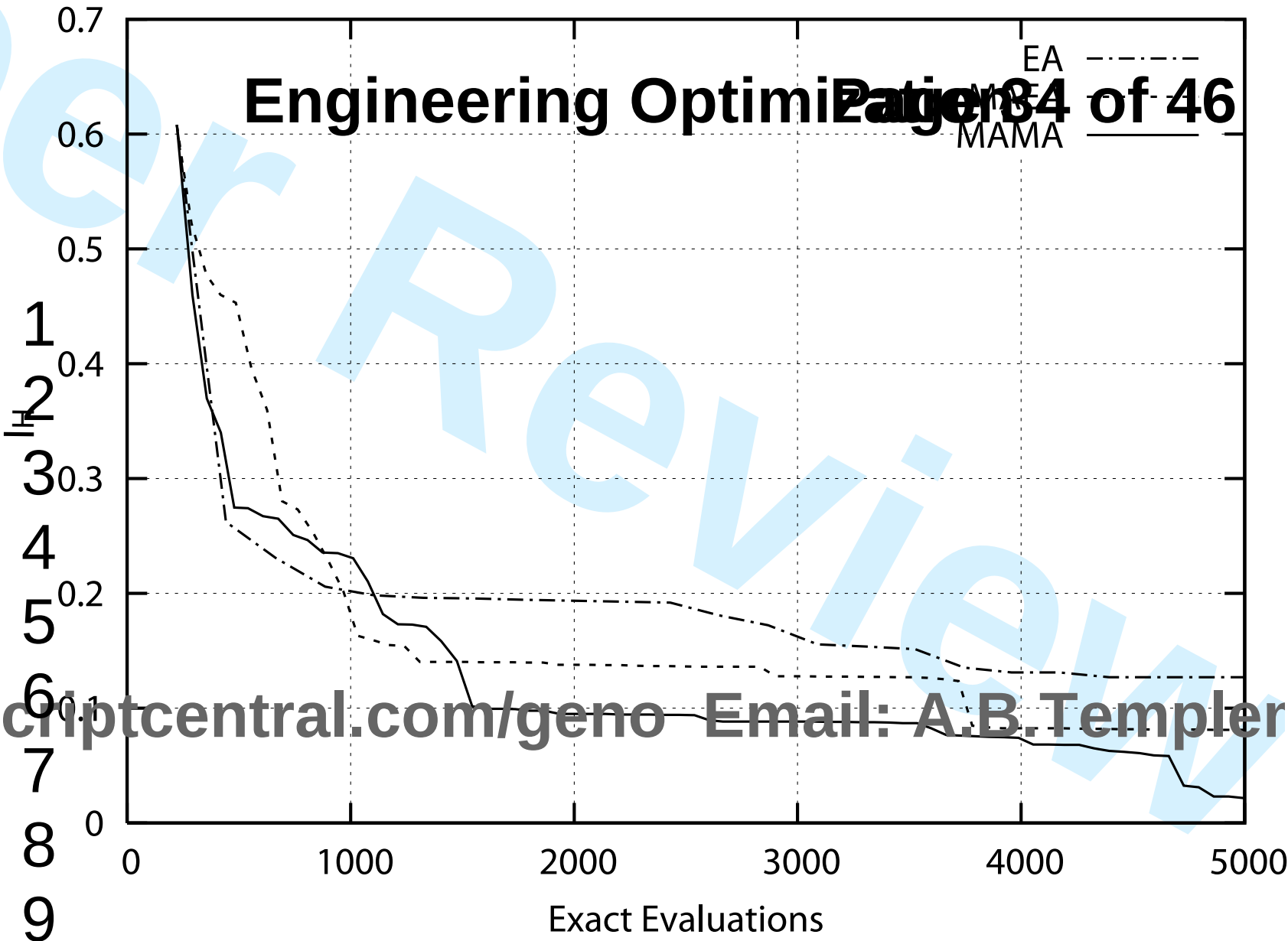


uscriptcentral.com/geno Email: A.B.Templem

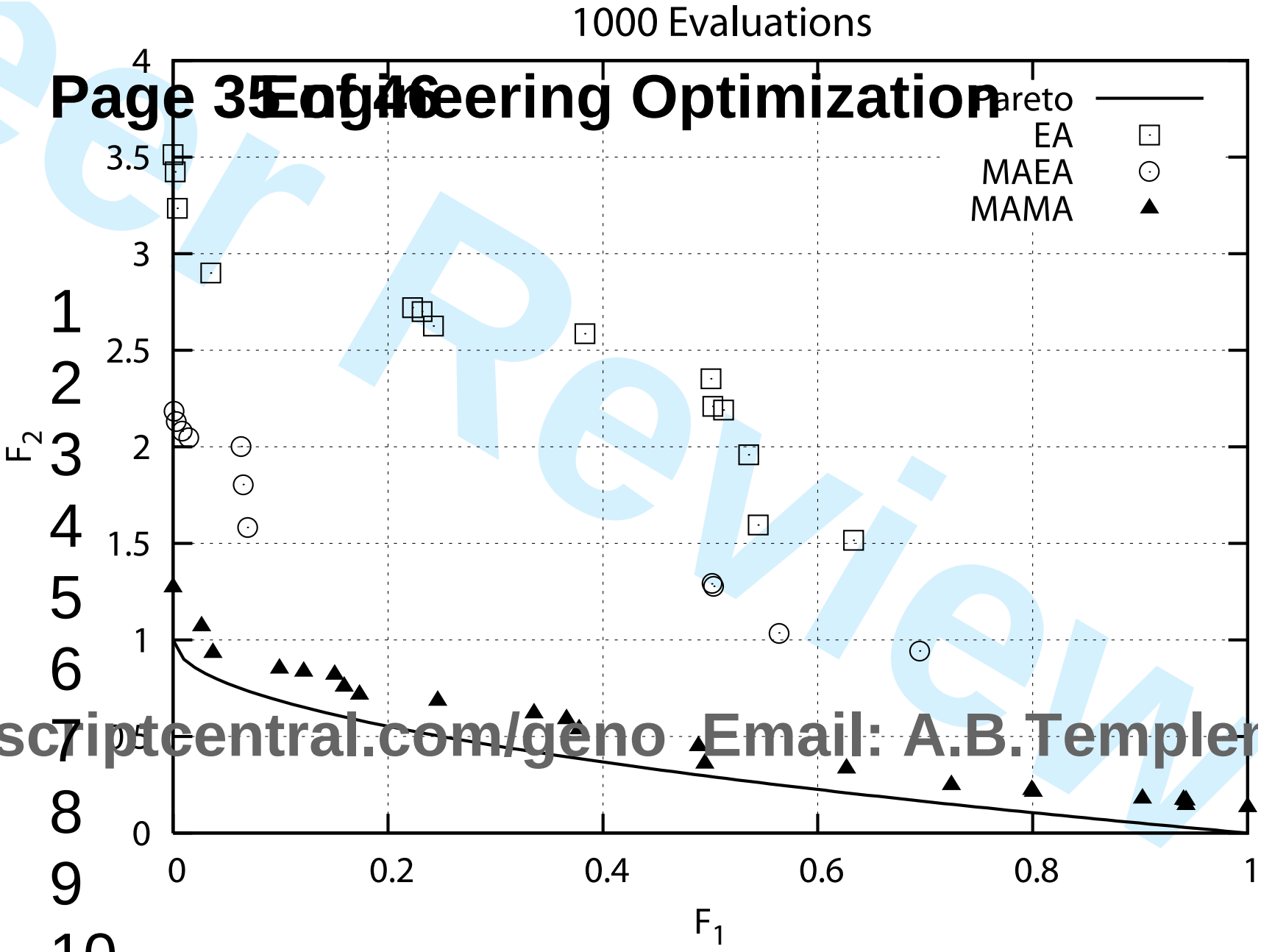


uscriptcentral.com/geno Email: A.B.Templem

For Peer Review

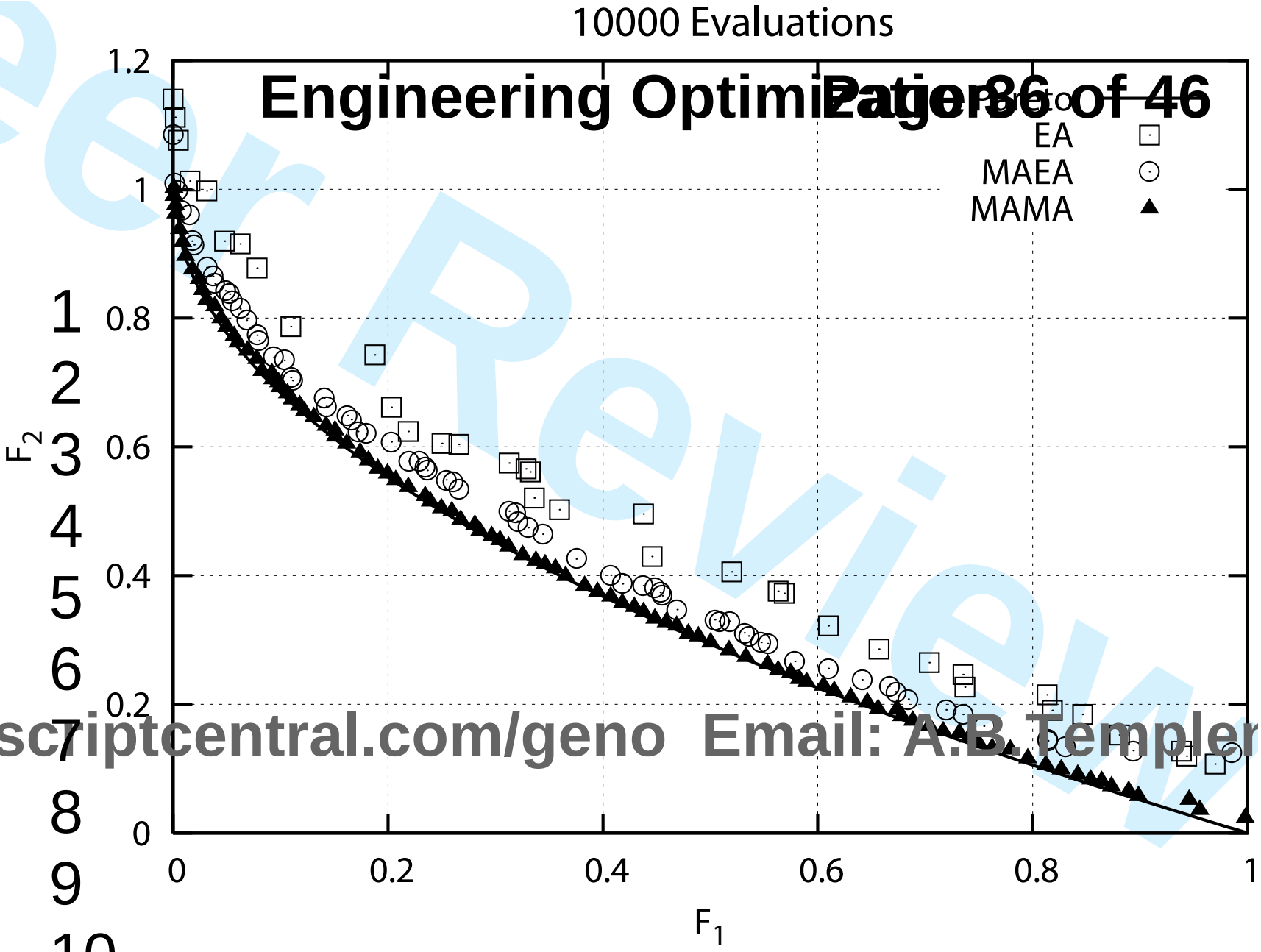


For Review
scriptcentral.com/geno Email: A.B.Templer

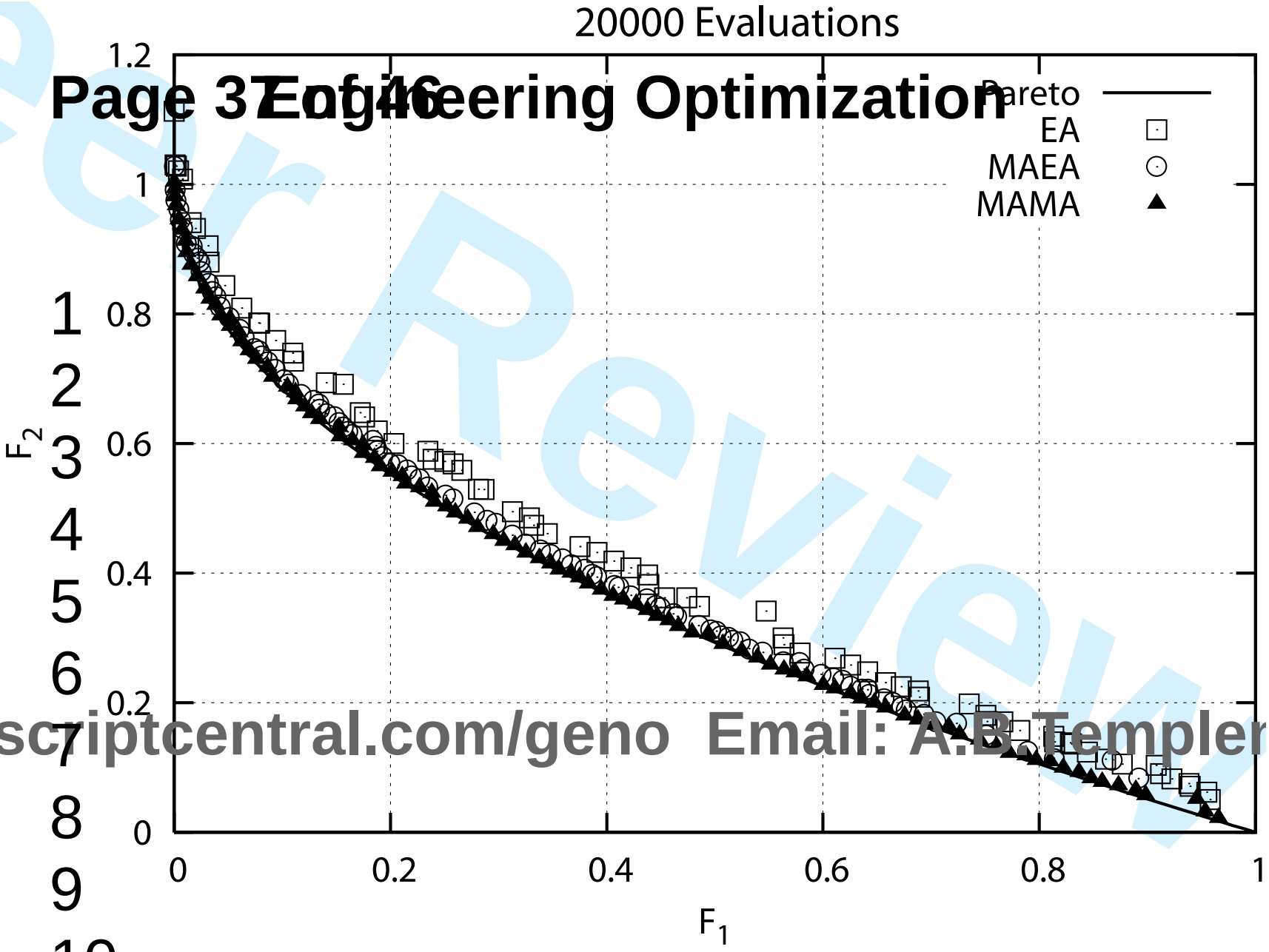


scriptcentral.com/geno Email: A.B.Templer

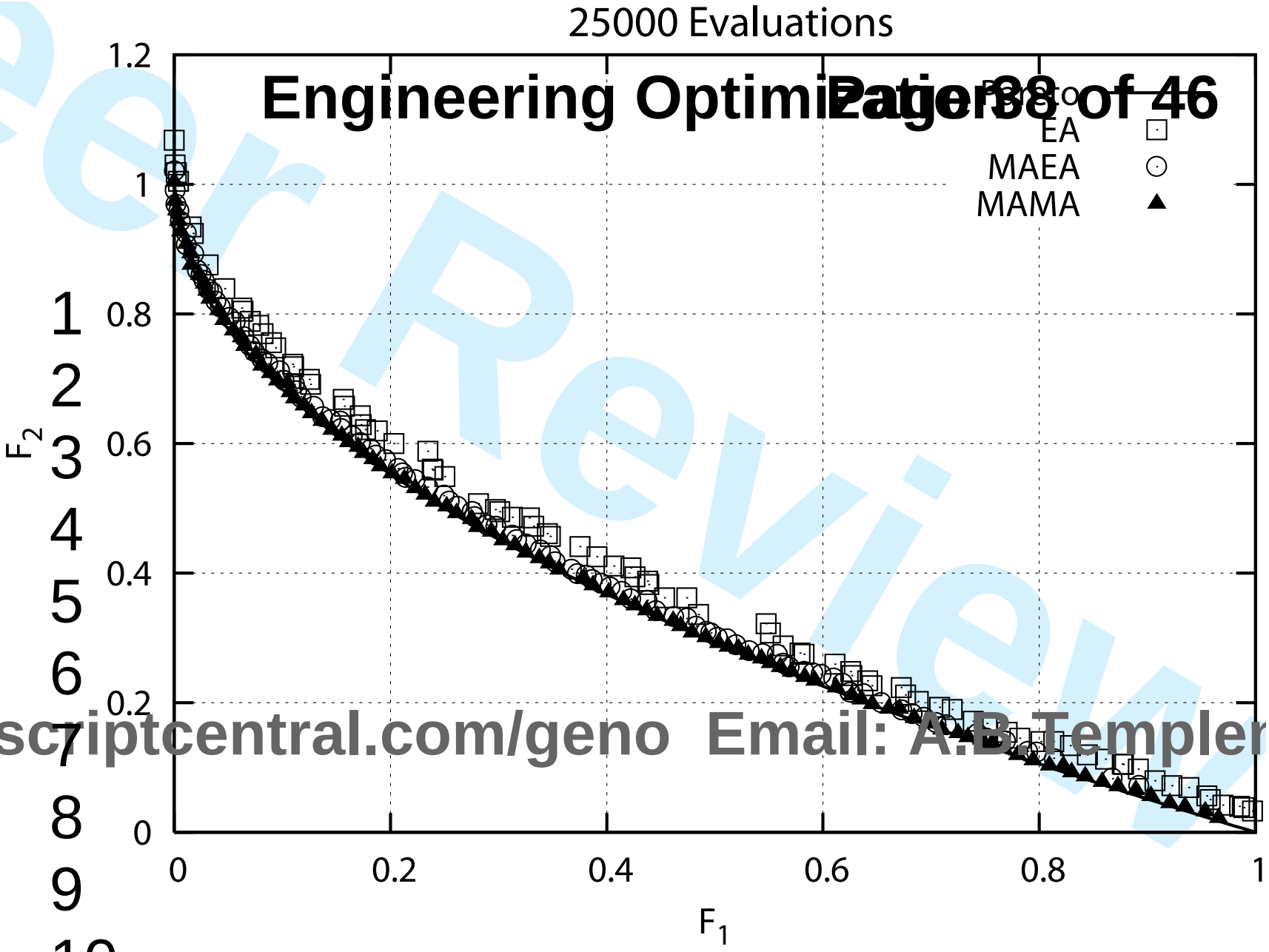
For Review



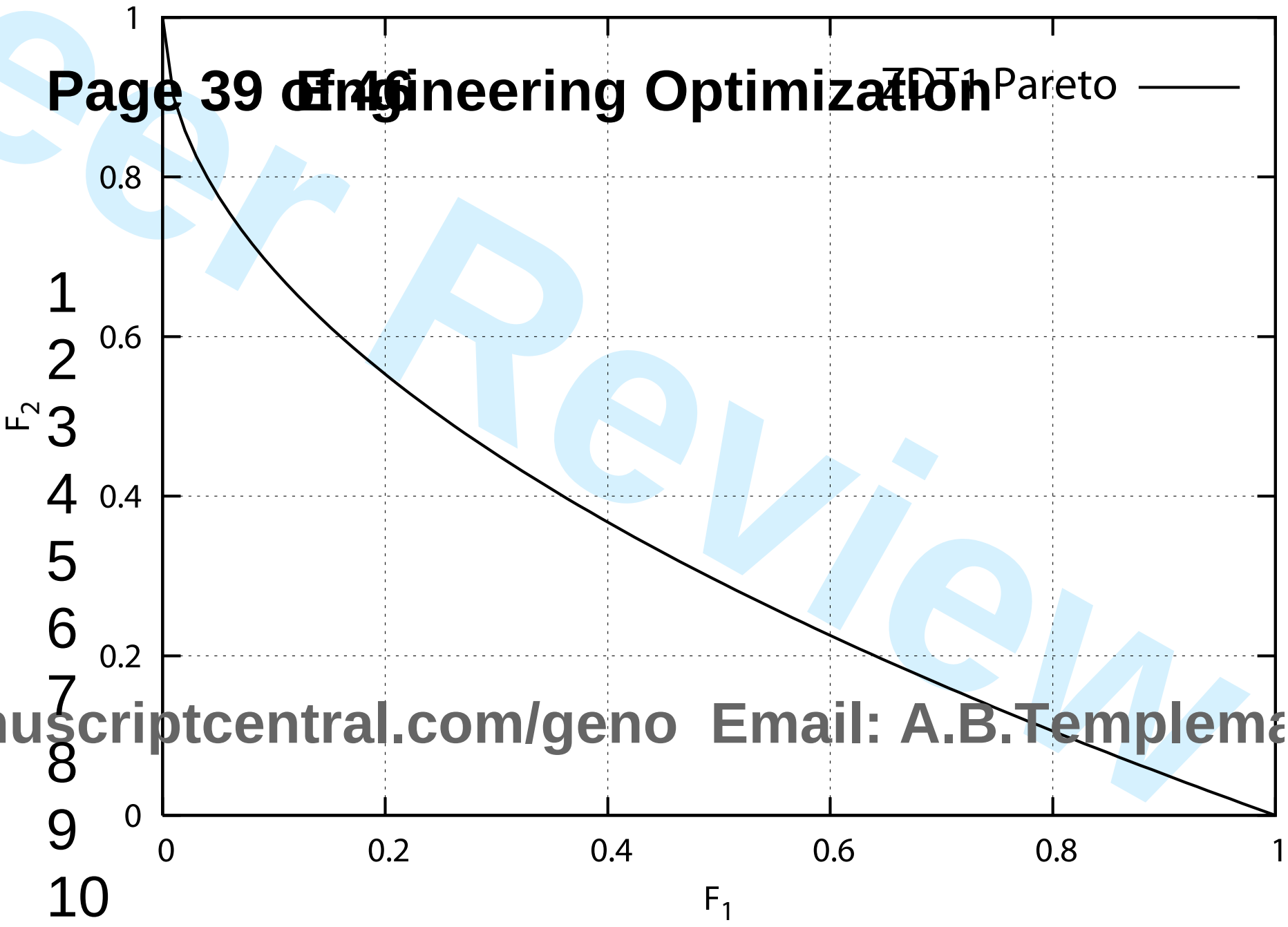
For Peer Review
scriptcentral.com/geno Email: A.B.Templer



For Peer Review
scriptcentral.com/geno Email: A.B.Templer

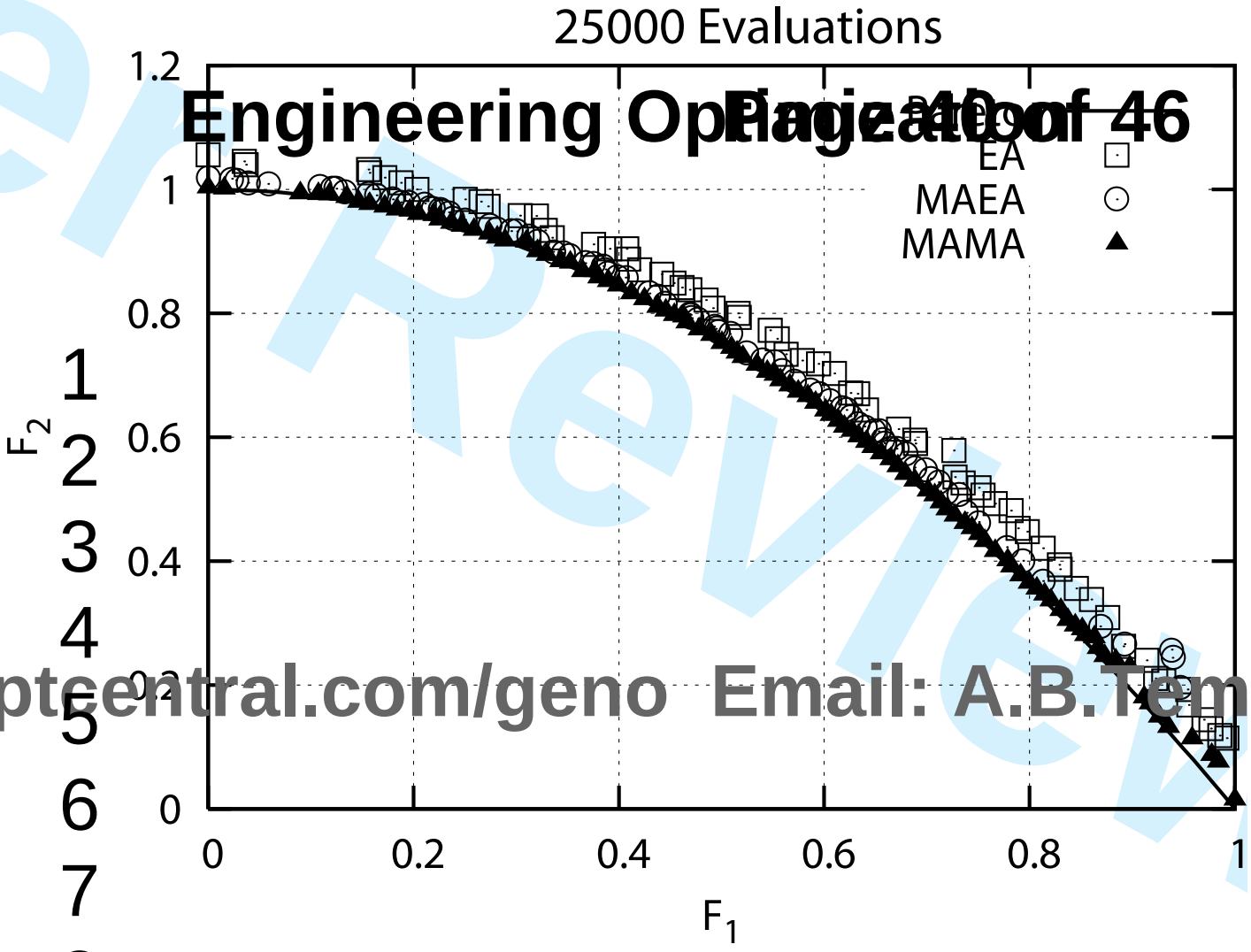


For Peer Review
scriptcentral.com/geno Email: A.B.Templer



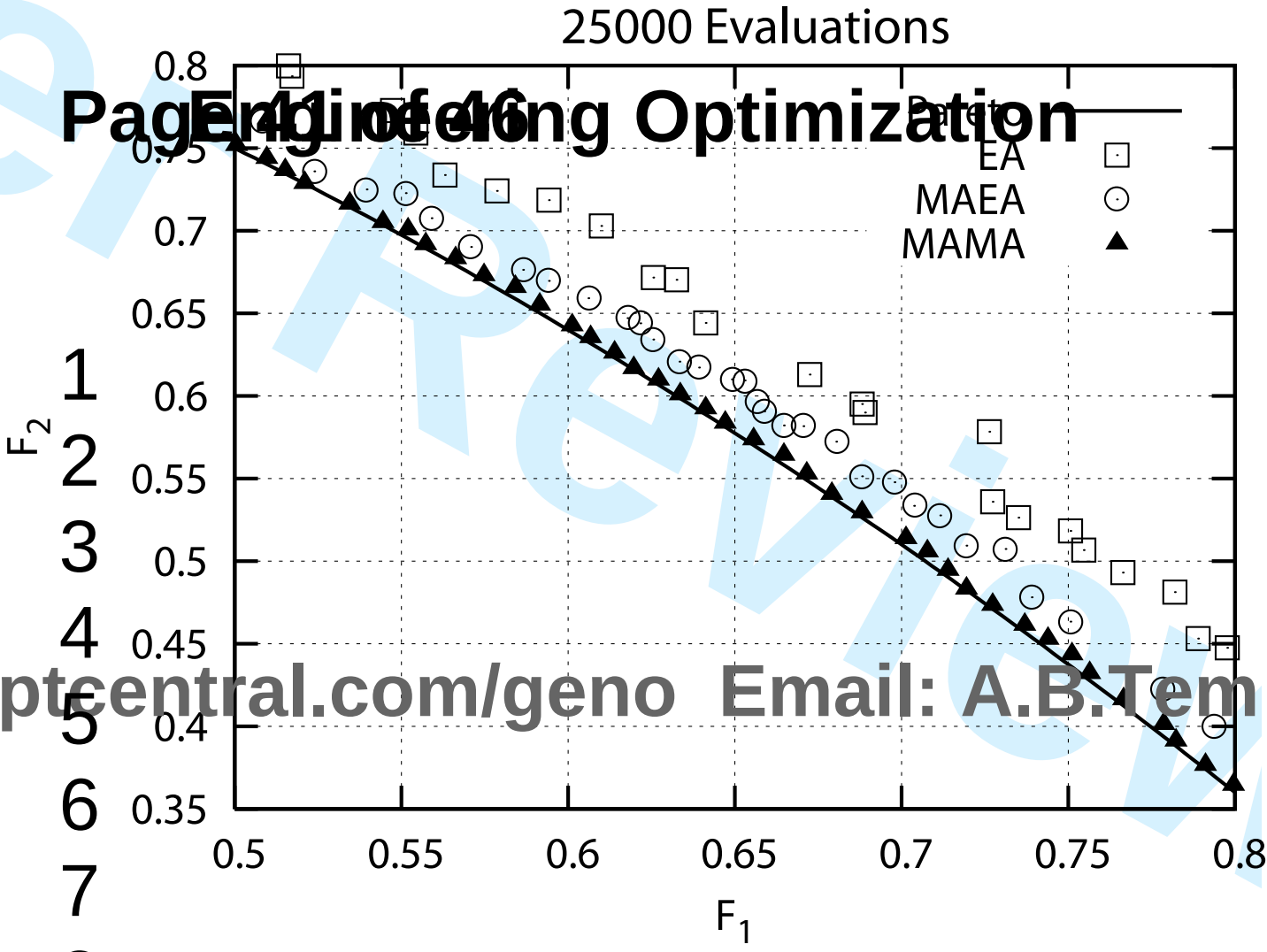
For Peer Review

nuscriptcentral.com/geno Email: A.B.Templema

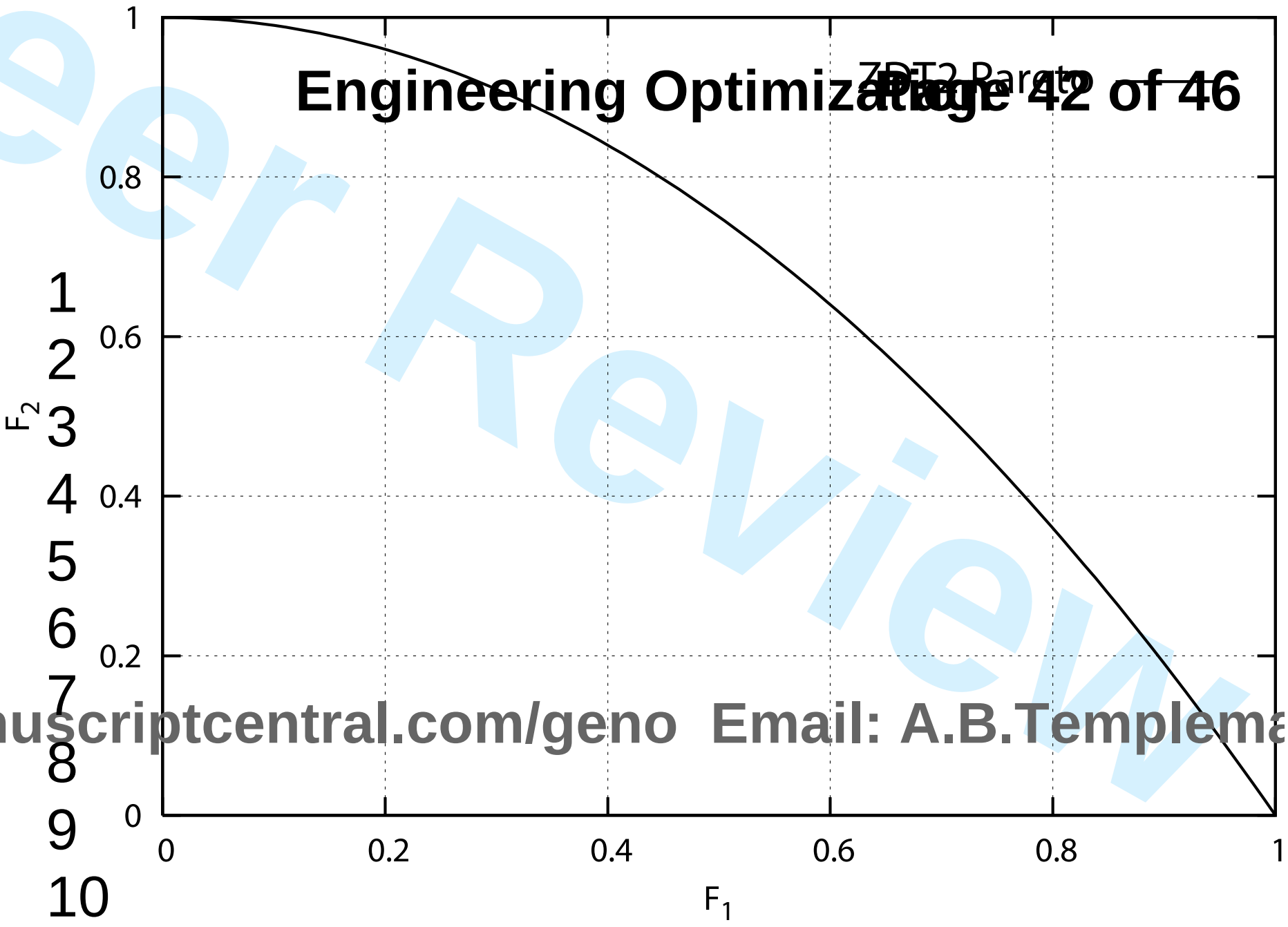


ptcentral.com/geno Email: A.B.Tem

Engineering Optimization

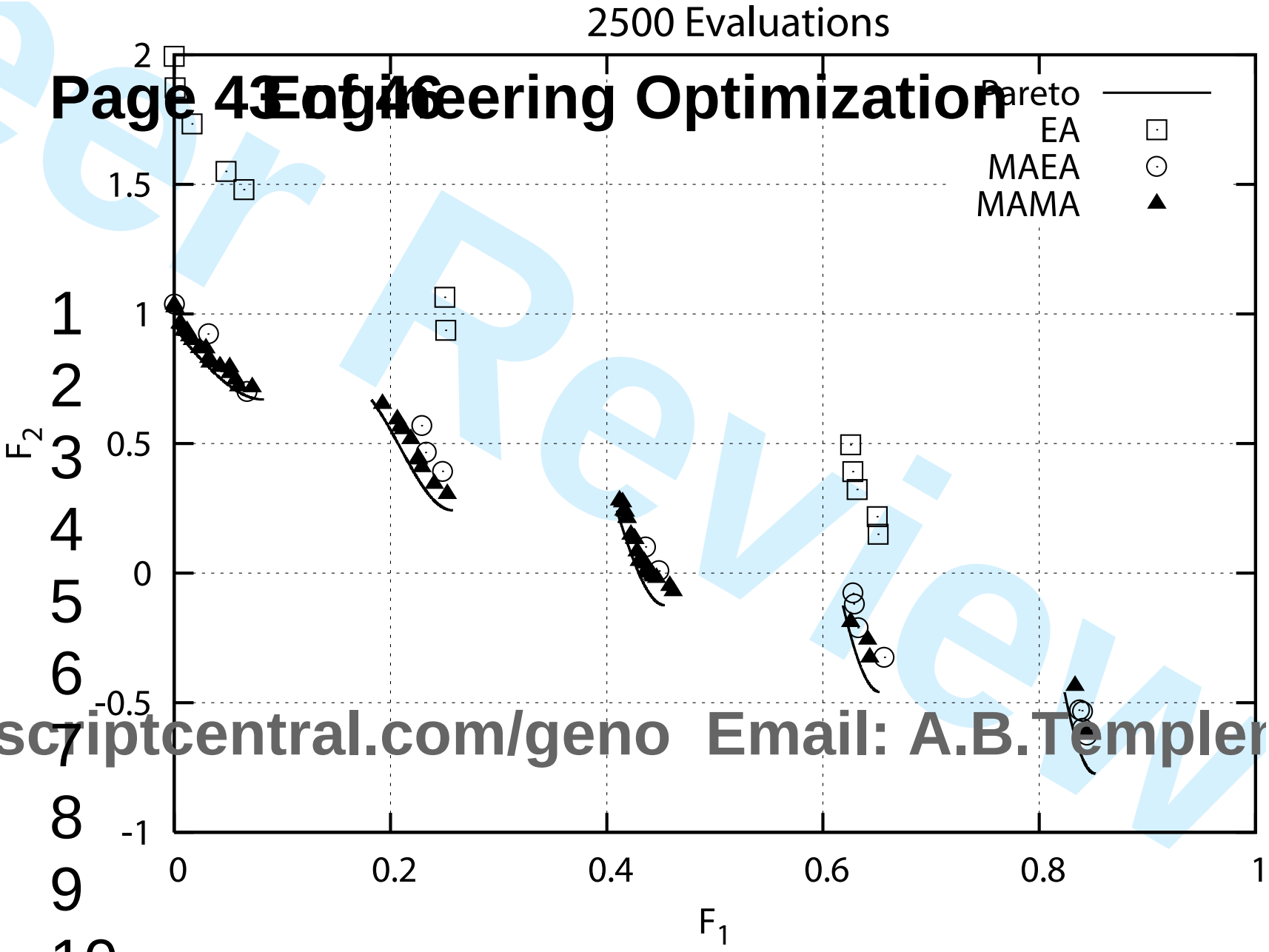


ptcentral.com/geno Email: A.B.Tem

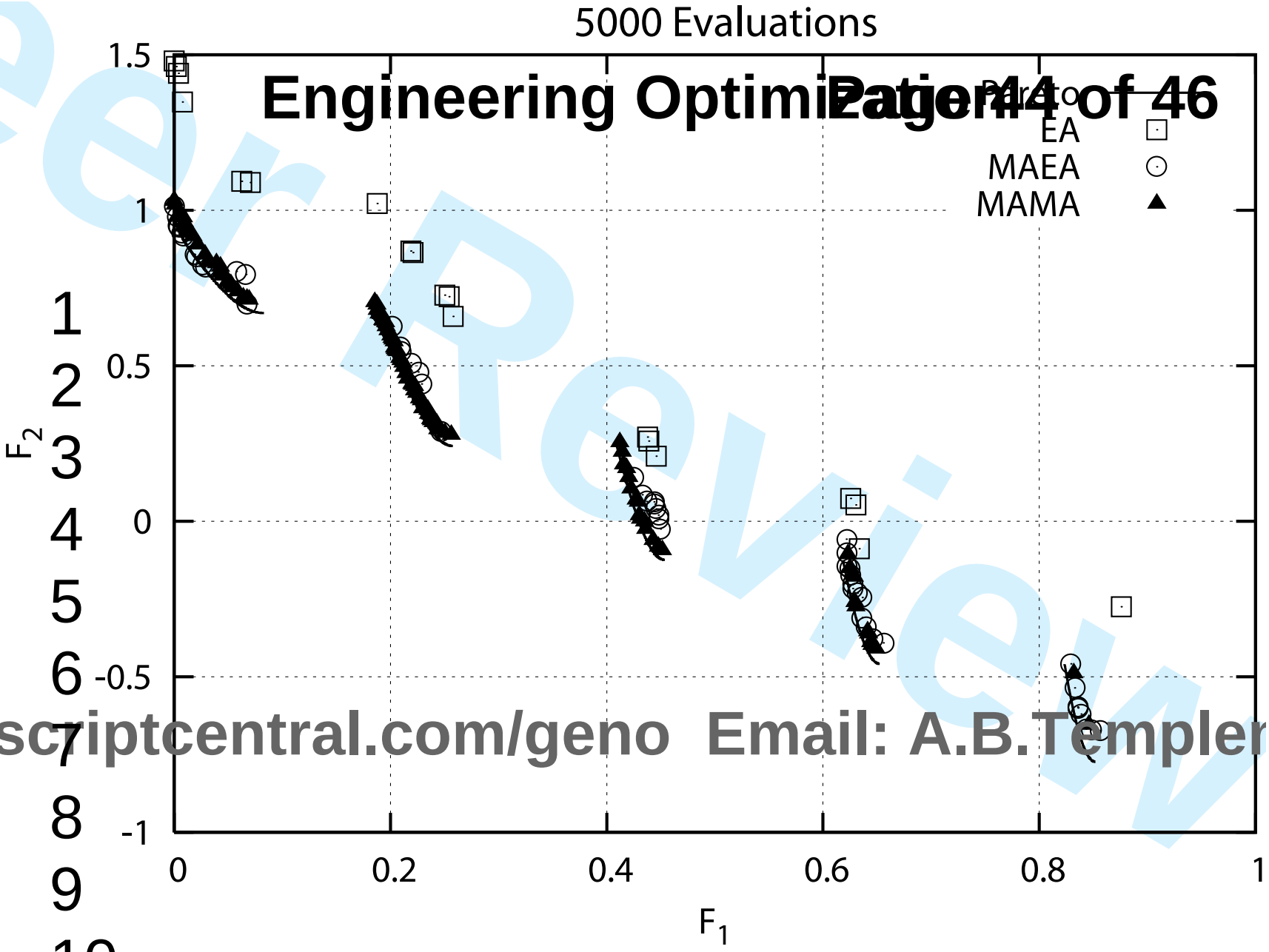


For Peer Review

nuscriptcentral.com/geno Email: A.B.Templema

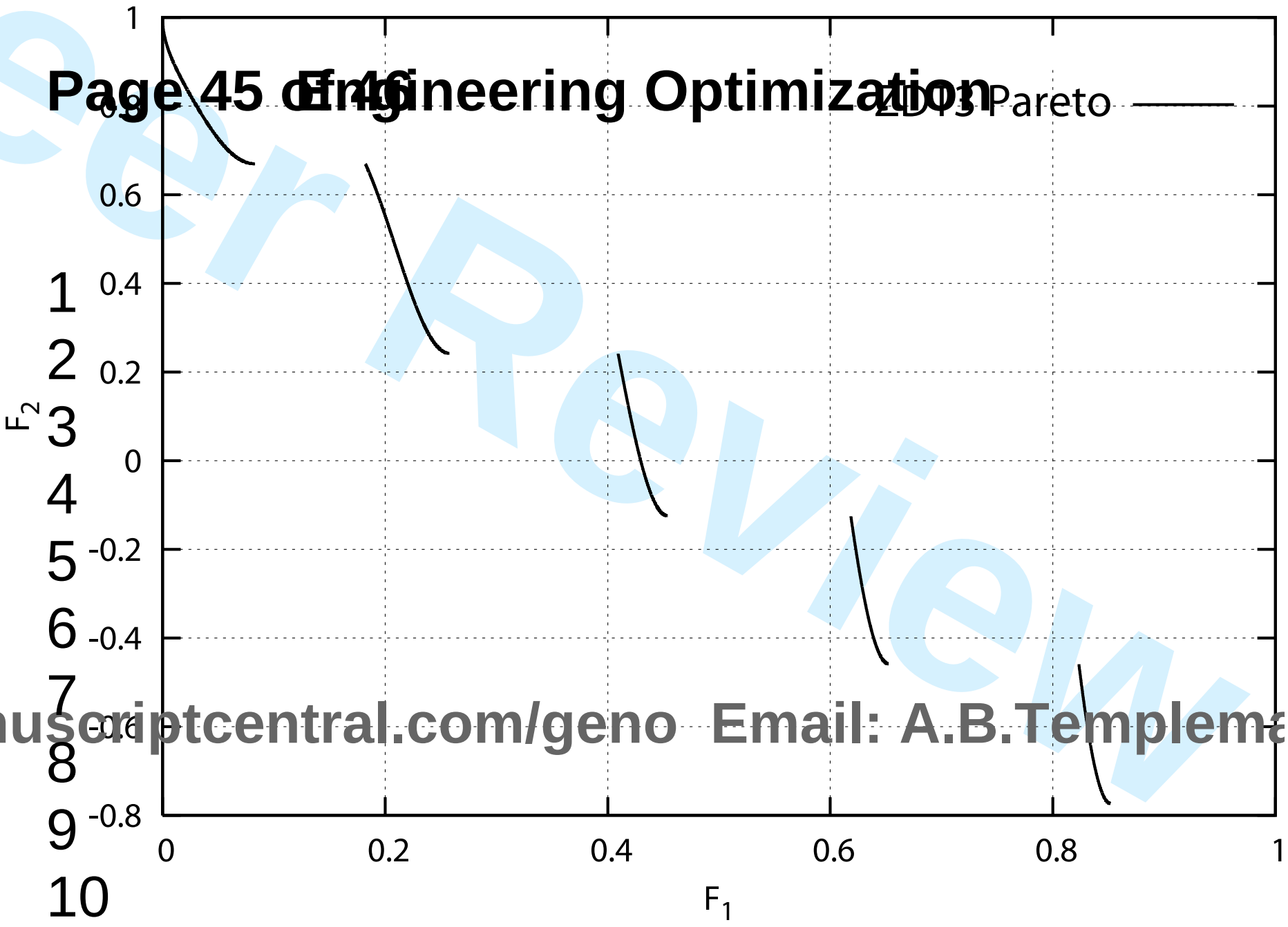


scriptcentral.com/geno Email: A.B.Templer



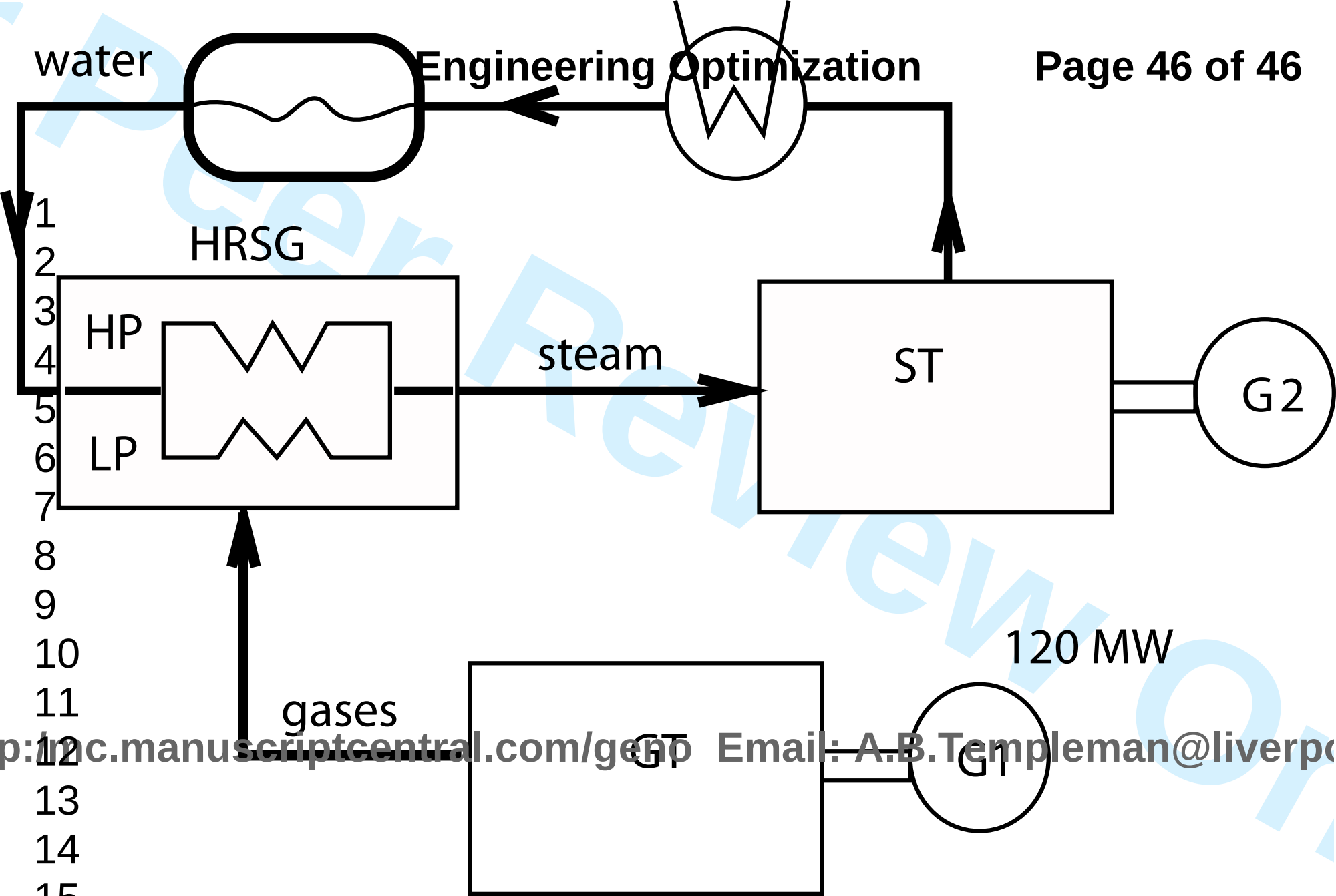
scriptcentral.com/geno Email: A.B.Templer

For Review



For Peer Review

nuscriptcentral.com/geno Email: A.B.Templema



http://mc.manuscriptcentral.com/geno Email: A.B.Templeman@liverpo

For Peer Review Only