21BR06A

Evaluation of sequential, multi-objective, and parallel interactive genetic algorithms for multi-objective optimization problems†

Alexandra Melike Brintrup,[1] Hideyuki Takagi,[2] Ashutosh Tiwari[1] and Jeremy J. Ramsden[1,*]

[1] *School of Applied Sciences, Cranfield University, Bedfordshire, MK43 0AL, UK*
[2] *Faculty of Design, Kyushu University, 4-9-1 Shiobaru, Minami-ku, Fukuoka 815-8540, Japan*

We propose a sequential interactive genetic algorithm (IGA), multi-objective IGA and parallel IGA, and evaluate them with both simulated and real users. Combining human evaluation with an optimization system for engineering design enables us to embed domain-specific knowledge that is frequently hard to describe, i.e. subjective criteria, and design preferences. We introduce a new IGA technique to extend the previously introduced sequential single objective GA and multi-objective GA, viz. parallel IGA. Experimental evaluation of three algorithms with a multi-objective manufacturing plant layout design task shows that the multi-objective IGA and the parallel IGA clearly provide better results than the sequential IGA, and that the multi-objective IGA gives the most diverse results and fastest convergence to a stable set of qualitatively optimum solutions, although the parallel IGA provides the best quantitative fitness convergence.

Keywords: innovative design, subjectivity, evolutionary computing

1. INTRODUCTION

Evolutionary Computation (EC) is the field of computational systems that use ideas and get inspiration from natural evolution [1]. Genetic Algorithms (GA) fall into the category of EC. GA are a type of search and optimization algorithm based on the mechanisms of genetics and natural selection.

The canonical form of GA encodes each candidate solution to a given problem as a binary, integer, or real-valued string, referred to as the chromosome. GA simulate the genetic evolution of a population of individuals using recombination operators such as crossover and mutation. Crossover exchanges genetic material between two parents during mating while mutation flips a bit in the chromosome, typically of the offspring. Mutation is carried out to prevent premature convergence of the design variables and promote diversity, which means preventing all the bit structures of strings in the mating pool from becoming identical in an early stage of the evolution. Each individual is evaluated once per generation according to some fitness criterion enabling a numerical fitness value to be assigned to the individual. New individuals are created by the recombination operators for the next generation.

It is important to realize that GA are stochastic, meaning that there is randomness involved; mainly in the initial generation of a random population, random choice of parents, random choice of which genes to inherit from which parents, and random choice of which genes to mutate. Sometimes however tournament selection is used as a parent or survival selection strategy. This ensures that there is a bias towards replacing less fit solutions in the parent population by fitter solutions from the new generation.

The algorithm is allowed to continue to produce new generations until a satisfactory solution is reached (or, for practical reasons, a preset number of generations is iterated). The procedure for a simple GA is illustrated in Figure 1.

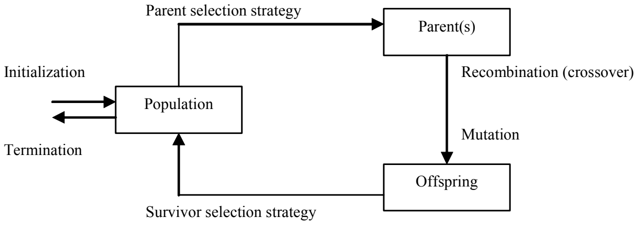
Parent selection strategy
Parent(s)
Initialization
Recombination (crossover)
Population
Mutation
Termination
Offspring
Survivor selection strategy

Figure 1. A simple genetic algorithm. One complete cycle constitutes one generation. Survivor selection strategy determines which offspring and which old parents are allowed to become parents or simply members of the next generation.

Interactive Evolutionary Computation (IEC) is an EC method that optimizes a target system based on subjective human evaluation of individuals, with the human playing the rôle of the fitness function in conventional EC [2]. When it is applied to fields that include a degree of subjectivity, such as engineering

† A preliminary version of this paper was circulated in F. Rothlauf et al. (Eds.): Evo Workshops 2006, LNCS 3907, pp.586-598, 2006.
* Corresponding author. Tel.: + 44 1234 754100. Fax + 44 1234 751346. E-mail: j.ramsden @cranfield.ac.uk

Received 14 August 2006; *accepted* 14 September 2006

design, the creation of art, the composition of music or architecture, interaction with a human evaluator helps the EC to generate solutions that incorporate his/her expertise or intuition without having to explicitly describe them in the optimization platform. Interaction between an IEC user and the computation can proceed in many ways. For instance, the user may participate in choosing élite designs for survival, modify an individual and reinsert it into the population of designs, or "freeze" parts of the design with the intention of reducing the search space dimensionality, besides the human fitness evaluation that is considered to be normal IEC. Parmee [3] has indeed redefined IEC broadly as system optimization based on human-machine interaction

Multi-objective design optimization is defined as the problem of finding a vector of decision variables that optimizes a vector function whose elements represent multiple objective functions. The concept of multi-objective optimization comes from the need to achieve compromise decision-making in problems incorporating many conflicting objectives. In such an environment, the ideal platform would enable us to gather a diverse set of solutions, each with its own offering of different levels of objective satisfaction, so that a choice of solution or solutions can easily be reached.

Our previous survey [4] showed that many problems previously believed to be quantitatively dominated, such as engineering design or system design, had (1) multiple conflicting objectives, and (2) subjective objectives among them. Subjective (qualitative) objectives may act as conflicting or complementary to quantitative objectives and are essentially unpredictable [5].

Recently, the use of EC techniques in multi-objective optimization has become popular. The process of solving a multi-objective optimization problem by EC is called Evolutionary Multi-objective Optimization (EMO). The particular suitability of EC for multi-objective optimization tasks together with the subjectivity inherent in many of those tasks led to the introduction of EMO incorporating IEC in order to capture subjective opinions of the designer [4]. Some other researchers followed similar ideas, such as gathering objective preferences from the designer [6], or asking the designer to pick favourable search areas [7], and our interactive multi-objective design optimization framework [8] used subjective opinions of the designer as an additional objective function, with promising results.

In this paper, we propose a new algorithm for handling multiple conflicting objectives including subjective (qualitative) objectives: the parallel interactive genetic algorithm (IGA) that optimizes each design objective in separate population islands before merging the solutions. The results of the parallel IGA are compared with two previously proposed algorithms, the sequential IGA and the multi-objective IGA based on Deb's non-dominated sorting genetic algorithm [9]. We describe the three IGAs in section 2, explain the experimental evaluation task and conditions in section 3, discuss the experiments in sections 4 and 5, and conclude the comparison of the three IGAs with an outline of the future work that remains to be done.

2. PLATFORMS DEVELOPED FOR INTERACTIVE MULTI-OBJECTIVE OPTIMIZATION

2.1 Sequential IGA

The sequential IGA acts as a single objective optimization platform, taking turns to optimize the quantitative and qualitative objectives independently.

A single population is evolved by a sequentially switched fitness function. Initially a subjective run is performed with a conventional IGA, meaning that the user needs to evaluate all the individuals of the current population displayed one design at a time, by giving them a qualitative rating between 0 and 9, 0 being the best design and 9 the worst. Users are allowed to give the same rating to more than one design. The subjective optimization is accomplished by the algorithm taking into account solely the user-given rating, which, to the algorithm, acts as a black-box fitness function. A subjective generation is then followed by a certain number of quantitative optimization runs in which fitness is evaluated by a regular fitness function.

This method treats the subjective and quantitative features of the design problem as separate objectives to be optimized. For instance, the individuals created from one qualitative run are fed into the following quantitative run as parent designs. This process ensures that the starting points of the quantitative runs are subjectively optimized designs. That is how the connexion between subjective and quantitative criteria is assured.

With this algorithm, we aimed to represent typical design cycles in an engineering design firm, where the design is sent back and forth between the marketing department, concerned with subjective aspects of the design, and the research and development (R&D) department, concerned with quantitative aspects of the design. Figure 2 shows the flow of sequential IGA in detail.

2.2 Multi-objective IGA

Multi-objective IGA is based on a modified version of a popular multiple objective optimization algorithm, version 2 of Deb's non-dominated sorting genetic algorithm [9].

The algorithm enhances the usual non-domination-based multi-objective optimization techniques by introducing

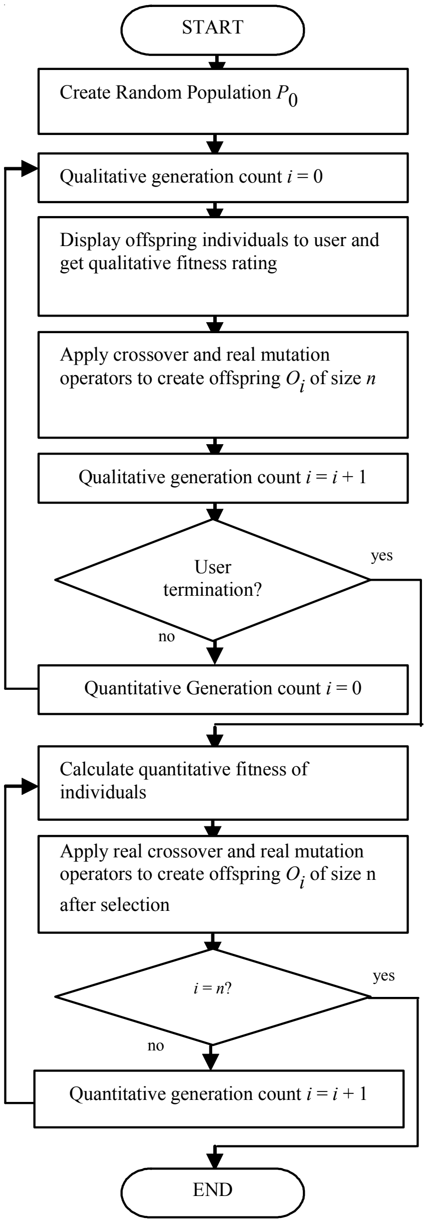
START
Create Random Population P_0
Qualitative generation count $i = 0$
Display offspring individuals to user and get qualitative fitness rating
Apply crossover and real mutation operators to create offspring O_i of size n
Qualitative generation count $i = i + 1$
User termination?
yes
no
Quantitative Generation count $i = 0$
Calculate quantitative fitness of individuals
Apply real crossover and real mutation operators to create offspring O_i of size n after selection
$i = n$?
yes
no
Quantitative generation count $i = i + 1$
END

Figure 2. Sequential IGA design optimization.

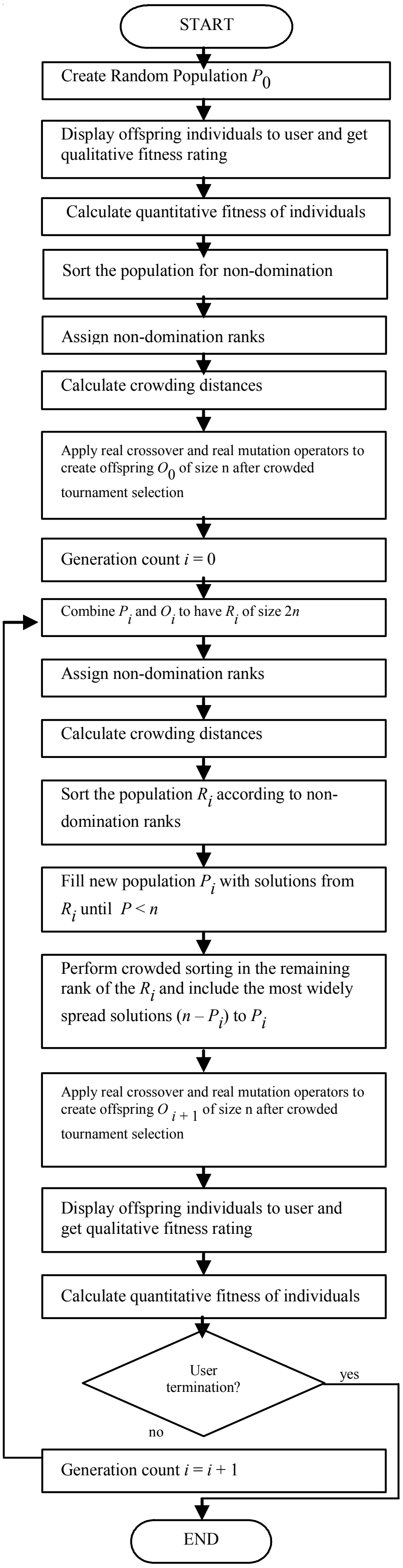
START
Create Random Population P_0
Display offspring individuals to user and get qualitative fitness rating
Calculate quantitative fitness of individuals
Sort the population for non-domination
Assign non-domination ranks
Calculate crowding distances
Apply real crossover and real mutation operators to create offspring O_0 of size n after crowded tournament selection
Generation count $i = 0$
Combine P_i and O_i to have R_i of size $2n$
Assign non-domination ranks
Calculate crowding distances
Sort the population R_i according to non-domination ranks
Fill new population P_i with solutions from R_i until $P < n$
Perform crowded sorting in the remaining rank of the R_i and include the most widely spread solutions $(n - P_i)$ to P_i
Apply real crossover and real mutation operators to create offspring O_{i+1} of size n after crowded tournament selection
Display offspring individuals to user and get qualitative fitness rating
Calculate quantitative fitness of individuals
User termination?
yes
no
Generation count $i = i + 1$
END

Figure 3. Multi-objective IGA.

the concepts of élitism and diversity. Élitism ensures that globally good solutions are preserved from generation to generation. In the non-dominated sorting algorithm GA-2, élitism is assured by combining parent and offspring populations before sorting them for non-domination. Non-domination of a solution illustrates how many solutions are better in all criteria than the current solution. The non-dominated GA-2 sorting also seeks to obtain solutions as diverse as possible on the Pareto front (i.e. the solution set observed when there are conflicting objectives, which contains solutions that are non-dominating relative to each other), by performing a crowding distance calculation (see ref. 9 for a detailed description of the crowded tournament operator and the overall non-dominated GA-2 sorting procedure).

The provision of diversity conservation mechanisms is important for the solution search not to become trapped in local optima. Diversity refers to the goal of finding a population of solutions that are not only optimal with respect to given criteria, but that also vary between each other, providing a better coverage of the search space, and promoting more informed and hence effective decision making.

We have modified Deb et al.'s GA [9] to include interactive fitness assessment and renamed it multi-objective IGA. In this algorithm a single population is optimized, as in sequential IGA. The qualitative objective value is gathered from the user for a single solution, whereas the quantitative fitness of a solution is assessed by the built-in fitness function. Figure 3 shows the flow of multi-objective IGA.

2.3 Parallel IGA

EC techniques are suitable for parallelization, as different crossover and mutation operators and different evaluation functions can be applied to different sets of individuals. It is possible to separate the individuals themselves to be evolved in different processors or programs, or to separate each location or program to perform different selection and recombination routines on each individual. In either case, the reason for choosing parallelization depends very much on the problem at hand. For instance, parallelization can be a solution for computationally demanding problems, in which selection and recombination routines are applied to individuals, or it can simply be a way to separate populations, as in our case. In either of these cases, three main parallelization techniques are widely used [10]:

- Master-slave parallelization, where a single processor maintains control over selection and uses the other processors only for crossover, mutation and evaluation of individuals. This is useful when there are few processors and very large evaluation times.
- The island model, where every processor runs an independent EC, using a separate subpopulation. The processors coöperate by regularly exchanging migrants (good individuals). This model is suitable for clustering populations.
- The diffusion model, where the individuals only mate with other individuals within the local neighbourhood. This approach is particularly suitable for massively parallel computers with a fast local intercommunication network.

The use of parallelization with EMO would appear to have the potential to be effective, as the goal of EMO is to find a set of good solutions rather than a single optimum. Nevertheless there are few published reports on this topic [11, 12], and the use of parallelization techniques in IEC has not been hitherto reported.

We hypothesize that the use of parallelization and interactivity together could be advantageous and natural for our problem. Since we deal with multiple conflicting objectives these could be evolved with separate populations, with élite migrants exchanged between them, i.e. following the island model. The subjective objective fitness of a solution could be obtained by user interaction and used to evolve one population, while the other population is evolved by a regular fitness function. Advantages of this method are anticipated to be that: (1) the quantitatively evolved population might emerge much faster; and (2) a compromise decision should be encouraged by the migration of élites between populations.

The features of our parallel IGA include:

1) Parallelization: our parallel IGA uses an island model and optimizes n separate populations with n separate objectives with migrants exchanged among them. In the experiments reported here, we used $n = 2$.

2) Migrant selection: the top three élite solutions are selected from each population for migration.

3) Replacement strategy: the worst three individual solutions from each population are replaced by the migrants

4) Migrant fitness assignment strategy:

In the population optimized using the quantitative objective, migrants are sorted with respect to their IGA (i.e. user-given) rating. If any two ratings are equal, sorting is done using the calculated quantitative objective fitness. After sorting, an arbitrary quantitative fitness is assigned to the migrants to ensure their survival, using the following procedure:

a) Sort immigrants according to qualitative fitness, in descending order, i.e. the individual with the best qualitative fitness has rank 1;

b) If any qualitative fitnesses are equal, sort them according to their quantitative fitnesses:

i) Assign the current generation's best quantitative fitness to the migrant with rank 1;

ii) Assign 110% of the current generation's best quantitative fitness to the migrant with rank 2;

iii) Assign 120% of the current generation's best quantitative fitness to the migrant with rank 3.

In the population optimized using the qualitative objective (goal), migrants are all given the minimum, i.e. the best, qualitative fitness. The reason is that the qualitative fitness rating is a discrete value and designs taking the same rating are allowed. However, the probability that any two designs would be assigned the same quantitative fitness after (computer) evaluation is very small (whereas even though two designs may differ from each other, the user might give them the same rating).

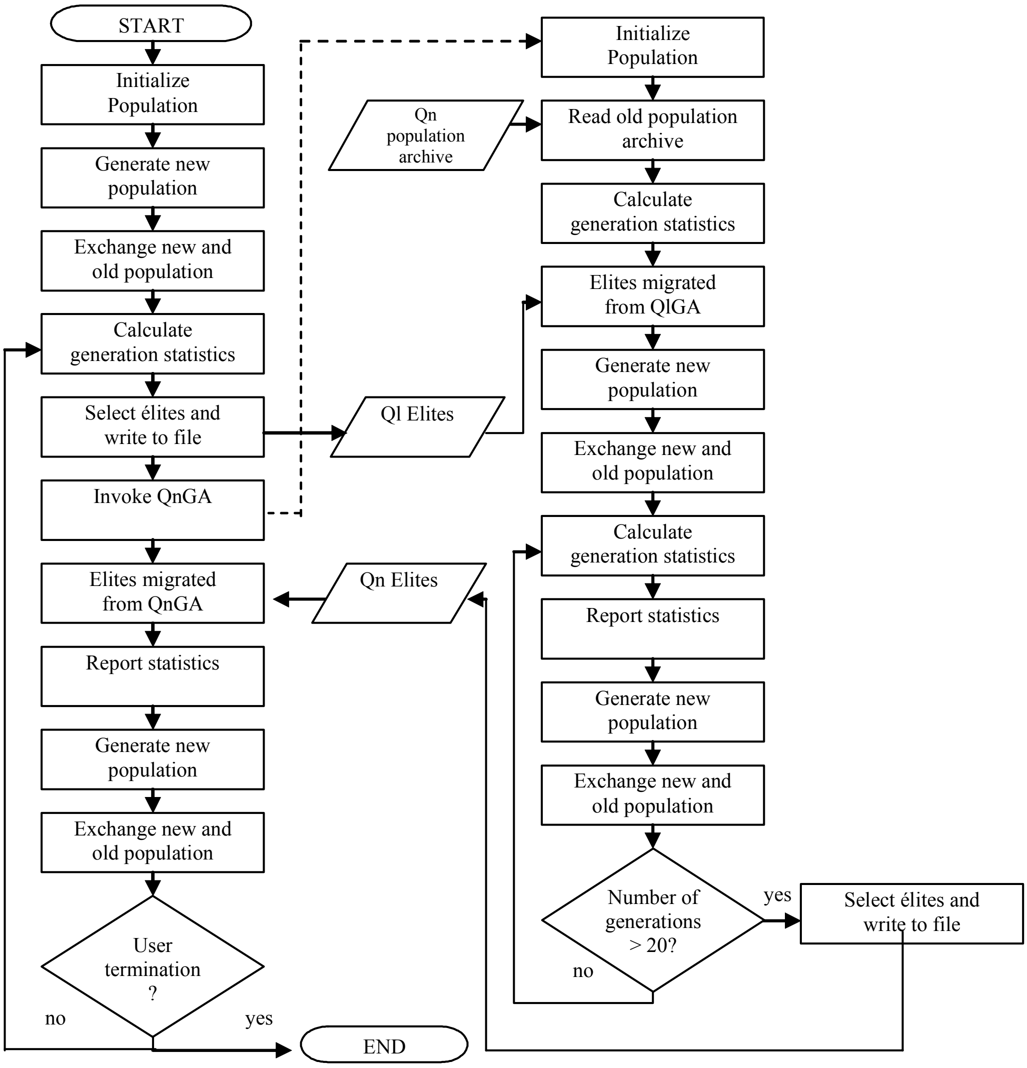
START
Initialize Population
Generate new population
Exchange new and old population
Calculate generation statistics
Select élites and write to file
Invoke QnGA
Elites migrated from QnGA
Report statistics
Generate new population
Exchange new and old population
User termination ?
no
yes
END
Initialize Population
Qn population archive
Read old population archive
Calculate generation statistics
Elites migrated from QlGA
Ql Elites
Generate new population
Exchange new and old population
Calculate generation statistics
Qn Elites
Report statistics
Generate new population
Exchange new and old population
Number of generations > 20?
yes
no
Select élites and write to file

Figure 4. Parallel IGA flow. Abbreviations: Ql, Qualitative; Qn, Quantitative ; QlGA, Qualitative Genetic Algorithm; QnGA, Quantitative Genetic Algorithm.

Two parallel IGA platforms are used for testing: pseudo parallel IGA that uses a pseudo IGA user as a fitness function, and real parallel IGA that is a normal IGA with a real human user. The user evaluation of the pseudo parallel IGA is simulated by a fitness function and there is no real user involvement. The pseudo parallel IGA was introduced to evaluate the maximum performance of the algorithm under an ideal condition without unpredictable human fickleness in the evaluation. Figure 4 shows the flow of the general parallel IGA procedure.

3. EXPERIMENTAL EVALUATION

3.1 Task description

Manufacturing plant layout design for an appliance manufacturing company was used as a benchmark task (evidently the task could equally well be considered to be the design for an experimental research institute, etc.). The objectives are to find the optimal width and length of each functional area, see Figure 5(a). An optimal choice will (1) minimize the building cost, which is directly related to the total area of the layout, and (2) maximize subjective user evaluation assigned to the interactive fitness evaluation element. This problem is an ideal candidate for testing the algorithms developed as it includes both quantitative and subjective (qualitative) features. Minimization of cost clearly constitutes a quantitative feature. On the other hand, preferences for the arrangement and sizes of the functional areas vary due to concerns such as: experts with different functions needing to be in close physical proximity to each other for knowledge exchange; certain functions having to be confined in the smallest possible areas in order to minimize cabling costs; certain pieces of equipment having to remain at a reasonable minimum safety distance from each other; and so on. These concerns are difficult and time-consuming to quantify and model as a fitness function. Hence their evaluation is ideal for outsourcing to the designer himself, who can interactively evaluate the design giving due regard to these concerns. Table 1 shows the parameters of the problem and how the overall area is deduced from these parameters. A more detailed description of this kind of problem can be found in ref. 4. Figure 5(b) shows an example of the user interface used in our experiments.

The users were asked to evaluate the design by paying special attention to the sizes of the press area and the warehouse. The greater the sizes of these areas, the greater the user satisfaction was presumed to be. The reasons for this included: (1) ensuring that the subjective and quantitative objectives be conflicting such that a Pareto front could indeed be reached and analysed; and (2) ensuring consistency between pseudo user evaluation as obtained from the pseudo parallel IGA and real user evaluation of the subjective components of the rest of the algorithms developed. Table 2 shows the fitness evaluation method for subjective and quantitative objectives in the relevant components of the three different algorithms.

Table 1. Manufacturing plant layout design problem parameters. The costs of the press and office areas are multiplied by two to reflect the greater building expense.

Room	Parameter	Parameter Label	Cost function
Stores area	width	X_0	$C_1=X_0\,(2.2-X_1)$
Press area	length	X_1	$C_2=2\,X_1\,X_2$
Press area	width	X_2	
Paint room	width	X_3	$C_3=X_3\,X_4$
Paint room	length	X_4	
Offices	length	X_5	$C_4=2[3.6-(X_0+X_6)]\,X_5$
Warehouse	width	X_6	$C_5=X_6\,X_7$
Warehouse	length	X_7	
Spares area	-	-	$C_6=X_1[3.6-(X_2+X_3)]$
Assembly area	-	-	$C_7=[3.6-(X_0+X_6)][2.2-(X_{1+}X_5)]+X_6[2.2-(X_1+X_7)]$

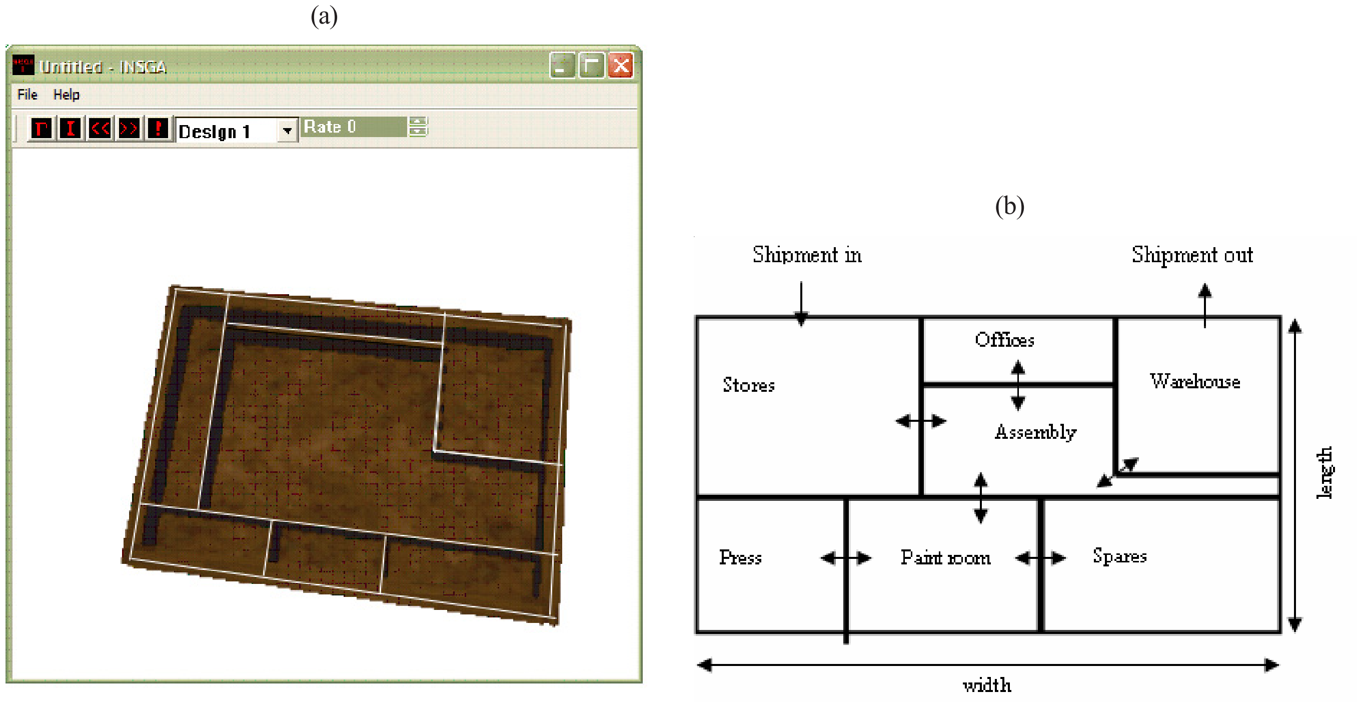

Figure 5. (a) Graphical user interface of the Multi-objective IGA platform , (b) Manufacturing plant layout design problem. Short double-headed arrows denote possible variation. The fixed overall width and length are 3.6 and 2.2 units respectively.

Table 2. Fitness evaluation in quantitative and subjective components of the parallel IGA, multi-objective IGA, and sequential IGA.

	Description	*Component*
Real human evaluation	A 10 point subjective rating is taken from the user.	Real parallel IGA subjective island. Multi-objective IGA subjective component. Sequential IGA subjective component.
$\sum_{i=1}^{7} C_i$	This function minimizes the total area of the rooms and thus reduces cost, which is directly proportional to the total area of the floor.	Pseudo parallel IGA quantitative island. Real parallel IGA quantitative island. Multi-objective IGA quantitative component. Sequential IGA quantitative component.
$\frac{1}{C_2 + C_4}$	This function simulates a user who requires bigger press and warehouse areas.	Pseudo parallel IGA subjective island.

3.2 Test parameters

Real GA coding, tournament selection, a mutation rate of 0.01, one-point simulated binary crossover with a rate of 0.9, distribution indices of 20 for simulated binary crossover and 10 for real mutation were used in both parallel IGA islands, sequential IGA and multi-objective IGA. GA properties such as recombination and selection operators, probabilities for selection, recombination and mutation, and population sizes were kept constant throughout to allow an accurate comparison of the three approaches. The human evaluators were 2 women and 3 men aged from 22 to 26 years. They included three product designers, one engineer, and one architectural engineer. The evaluators continued to run the program until subjective generation 5 was reached. Each evaluator conducted one test for each of parallel IGA, sequential IGA, and multi-objective IGA. For sequential IGA, with each evaluator 6 runs were performed, 3 of which were subjective and 3 of which were quantitative runs. Each subjective run consisted of 5 generations, while each quantitative run consisted of 10 generations. For multi-objective IGA, one run of 5 generations was carried out with each user. For the parallel IGA, the numbers of generations and the population sizes for each island are given in Table 3.

Table 3. Parallel IGA parameters for the quantitative and subjective population islands.

	Population size	Number of generations at each run	Total number of generations	Total number of user evaluations
Parallel IGA - Quantitatively optimized population island	50	10	50	0
Parallel IGA - Subjectively optimized population island	12	1	5	60
Multi-objective IGA	12	1	5	60
Sequential IGA - Quantitative optimization run	12	10	50	0
Sequential IGA - Subjective optimization run	12	1	5	60

4. RESULTS

Sequential IGA, multi-objective IGA, and Parallel IGA were compared over 55, 5, and 55 generations respectively, in terms of overall average subjective fitness, overall average quantitative fitness, average subjective fitness of the last generation, and average quantitative fitness of the last generation. The Wilcoxon signed rank test, a nonparametric pair observation test, was used to compare the results from the three algorithms.

Figure 6 shows the average fitness values obtained at each generation of each run. In the sequential IGA the qualitative average fitness showed a slightly worsening trend in the five runs pursued, while the quantitative results showed a result that was smoothly improving. In the multi-objective IGA, during a single iteration, two objective values were obtained for each design, as opposed to one objective value in the case of the sequential IGA. The qualitative and quantitative fitness

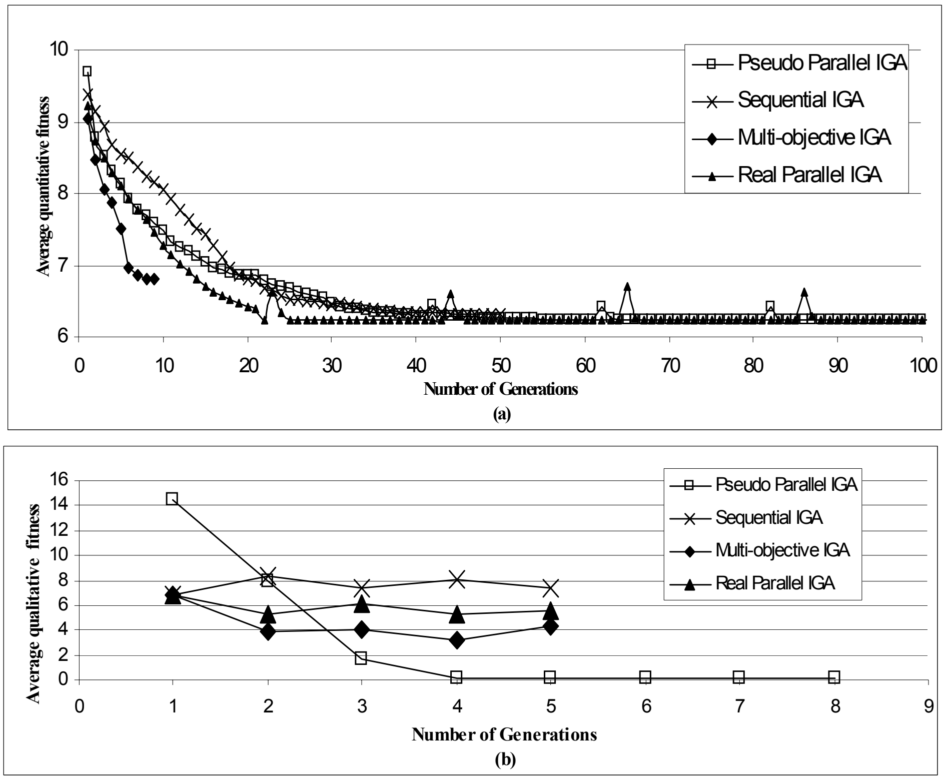

Figure 6. Changes of average (a) quantitative and (b) qualitative fitness in sequential IGA, multi-objective IGA, pseudo IGA, and real parallel IGA. The small peaks are due to the arrival of subjectively élite migrants in the pseudo and real parallel IGAs.

averages both showed an improving trend, and convergence to the Pareto front was observed. As the number of generations increased the solutions were minimized with respect to both criteria. For the parallel IGA, the Wilcoxon test indicates that:

- The pseudo parallel IGA is significantly advantageous over both the sequential IGA and the multi-objective IGA in terms of overall average subjective fitness, overall average quantitative fitness, average subjective fitness of the last generation, and average quantitative fitness of the last generation, with a risk of 0.05.
- The real parallel IGA is significantly advantageous over the multi-objective IGA in terms of overall average subjective fitness, overall average quantitative fitness, and average quantitative fitness of the last generation, with a risk of 0.05. The significance of the average subjective fitness of the last generation could not be evaluated with the relatively small number of tests carried out.
- The real parallel IGA is significantly advantageous over the sequential IGA in terms of overall average subjective fitness, overall average qualitative fitness, and average quantitative fitness of the last generation, with a risk of 0.05. The significance of the average quantitative fitness of the last generation could not be evaluated with the relatively small number of tests performed.

5. DISCUSSION

In multi-objective problems, it is important that a set of good solutions that are diverse from each other are obtained, so that compromise decision-making can be implemented. This section discusses the results in terms of fitness convergence and diversity of results, after laying out the main sources of error. We assume that the solution is quantitatively stable after about 40 generations.

5.1 Sources of error

Pseudo parallel IGA showed significantly better fitness convergence in both subjective and quantitative objective values than the multi-objective IGA and the sequential IGA. However, the multi-objective IGA and the sequential IGA involved real human evaluation whereas the parallel IGA did not. In order to mimic the fitness function in parallel IGA, users were asked to evaluate designs by observing the sizes of the warehouse and press area during user evaluation in the multi-objective IGA and the sequential IGA. However, noise due to human inconsistency is inevitable and should be taken into account in the assessment of results.

Another point to bear in mind while assessing these results is the number of generations pursued with each algorithm. The multi-objective IGA pursues one quantitative and one subjective generation together at a time since the evolution is simultaneous. As the user must be involved in every generation and his/her fatigue must be considered, only 5 quantitative generations could be evolved as opposed to 55 in the sequential IGA and the parallel IGA. We observed that even with only five generations, the performance of multi-objective IGA in both objective values was comparable to that of the parallel IGA and the sequential IGA.

5.2 Fitness convergence

The multi-objective IGA and parallel IGA reached satisfactory designs in an acceptably short time of 4 to 5 generations, whereas the sequential IGA failed to reach an equally satisfactory design in the opinion of the users. Although the final fitness scores of the quantitative objectives were better in the sequential and parallel IGAs than in the multi-objective IGA, since the subjective objective was of equal importance to the design overall, the sequential IGA performed significantly worse than the other two IGAs. Rather than reaching a compromise decision between qualitative and quantitative factors, the sequential IGA makes the two objectives compete against each other, resetting itself and trying to recover from the effects of the opposite objective each time. On the other hand, no significant difference was found between the averages from the parallel and multi-objective IGAs at the final generation, although the overall fitness average achieved by the parallel IGA is better than that achieved by the multi-objective IGA.

5.3 Diversity

After the second quantitative run of the sequential IGA, the quantitative objective took over, providing designs with little or no difference between them for qualitative evaluation by the user. The users reported difficulty in distinguishing the designs, even though minor differences still existed, but their visualizations were difficult and sometimes differences were barely apparent. This led the users to give similar ratings to designs, and it became difficult for the algorithm to diversify them. On the other hand, the diversity preservation mechanism, with the help of the crowding distance calculation in the multi-objective IGA, provided results that were visually distinct from each other. The population that showed the most variation between its members was the multi-objective IGA population, followed by that of the parallel IGA, and then that of the sequential IGA.

6. CONCLUSIONS

This paper compares three IGA algorithms, namely a novel parallel IGA, multi-objective IGA and sequential IGA that have been developed to optimize conflicting subjective and quantitative multiple objectives. The algorithms were evaluated with a problem of manufacturing plant layout design.

The major advantage of parallel IGA is its flexibility to accommodate more than one population. The population size of multi-objective IGA has to be constant as multiple objectives are dealt with simultaneously in each design, while that of the parallel IGA is adjustable according to human limitations in the subjectively evolved designs and according to the potential of the computer in the quantitatively evolved designs. Hence the time spent by the human user during design evaluation is better utilized, and as more generations can be evolved in the quantitative objective side, we can get a better fitness in this objective space. As there exist different populations, the parallel IGA does not engage in a fight between two contradictory objectives, as occurred with sequential IGA. Thus we can conclude that both the multi-objective IGA and the parallel IGA are significantly better than the sequential IGA and that sequential optimization cannot give satisfactory results in dealing with multiple objectives in conflict. The parallel IGA is observed to be rather satisfactory for the incorporation of multiple criteria principles even though in itself the algorithm is not a multi-objective optimization algorithm. In dealing with conflicting subjective and quantitative design objectives, both multi-objective IGA and the parallel IGA seem to be promising approaches.

Although the quantitative objective remains implicit in the multi-objective IGA, the parallel IGA displays to the user the designs that emigrated from the quantitative objective island to the subjective objective island. Therefore, in addition to the above advantage, real parallel IGA can help promote innovative decision making by making the user observe computer generated results and reconsider the evaluations.

Future work will focus on investigating the performance of parallel IGAs in comparison with other multi-objective optimization algorithms and with different benchmark problems. Additionally, a pairwise preference experiment is to be undertaken for the final populations achieved with the help of the two algorithms.

ACKNOWLEDGMENTS

This work was supported in part by the 2005 Summer Programme of the Japan Society for the Promotion of Science (JSPS) and a Travel Grant from the Collegium Basilea (Institute of Advanced Study).

REFERENCES

1. Banzhaf, W., Beslon, G., Christensen, S., Foster, J.A., Kepes, F., Lefort, V., Miller, J.F., Radman, M., Ramsden, J.J., From evolution to computational evolution: a research agenda. *Nature Review Genetics* **7** (2006) 729–735.
2. Takagi, H., Interactive evolutionary computation: fusion of the capacities of EC optimization and human evaluation. *Proceedings of the IEEE* **89** (2001) 1275–1296.
3. Parmee, I.C. Poor-definition, uncertainty and human factors—a case for interactive evolutionary problem reformulation? In: *Proc. Genetic Evolutionary Computing Conference*, Chicago, USA (July 2003).
4. Brintrup, A., Ramsden, J. & Tiwari, A. Integrated qualitativeness in design by multi-objective optimization and interactive evolutionary computation In: *Proc. IEEE Congress on Evolutionary Computation*, pp. 2154–2160. Edinburgh, UK (September 2005).
5. Bain, A. *The Senses and the Intellect*, p. 3. London: Parker (1855).
6. Parmee, I. C., Cvetkovic, D., Watson, A. & Bonham, C. Multi-objective satisfaction within an interactive evolutionary design environment. *J. Evolutionary Computation* **8** (2000) 197–222.
7. Kamalian, R., Takagi, H. & Agogino, A. Optimized design of MEMS by evolutionary multi-objective optimization with interactive evolutionary computation. *Genetic and Evolutionary Computation Conference*, Seattle, USA (2004) 1030–1041.
8. Brintrup, A., Tiwari, A. & Gao, J. Handling qualitativeness in evolutionary multiple objective engineering design optimization. *Enformatica* **1** (2004) 236–240.
9. Deb, K., Agrawal, S., Pratap, A. & Meyarivan, T. A fast élitist non dominated sorting genetic algorithm for multi objective optimization: NSGA-2, *Proc. Parallel Problem Solving from Nature*, Paris, France (2000) 858–862.
10. Cantu-Paz, E. *Efficient and Accurate Parallel Genetic Algorithms*. Norwell, Kluwer Academic (2000).
11. Van Veldhuizen, D.A., Zydallis, J.B. & Lamont, G.B. Considerations in engineering parallel multiobjective evolutionary algorithms. *IEEE Trans. Evolutionary Computation* **7** (2003) 144–173.
12. Hiroyasu, T., Miki, M. & Watanabe, S. The new model of parallel genetic algorithm in multiobjective optimization problems -divided range multi-objective genetic algorithms. In: *Proc. IEEE Congress on Evolutionary Computation*, pp. 333–340. La Jolla, California (July 2000).