

A Technique for Evaluation of Interactive Evolutionary Systems

M. Shackelford¹, D. W. Corne²

¹School of Systems Engineering,
University of Reading,
Reading RG6 6AY
Mark.Shackelford@inkfish.co.uk

²School of Engineering, Computer Science and Mathematics
University of Exeter
Exeter EX4 4QF
D.W.Corne@exeter.ac.uk

Abstract

Very large scale scheduling and planning tasks cannot be effectively addressed by fully automated schedule optimisation systems, since many key factors which govern ‘fitness’ in such cases are unformalisable. This raises the question of an interactive (or collaborative) approach, where fitness is assigned by the expert user. Though well-researched in the domains of interactively evolved art and music, this method is as yet rarely used in logistics. This paper concerns a difficulty shared by all interactive evolutionary systems (IESs), but especially those used for logistics or design problems. The difficulty is that *objective evaluation* of IESs is severely hampered by the need for expert humans in the loop. This makes it effectively impossible to, for example, determine with statistical confidence any ranking among a decent number of configurations for the parameters and strategy choices. We make headway into this difficulty with an *Automated Tester* (AT) for such systems. The AT replaces the human in experiments, and has parameters controlling its decision-making accuracy (modelling human error) and a built-in notion of a target solution which may typically be at odds with the solution which is optimal in terms of formalisable fitness. Using the AT, plausible evaluations of alternative designs for the IES can be done, allowing for (and examining the effects of) different levels of user error. We describe such an AT for evaluating an IES for very large scale planning.

1 Introduction

Large-scale optimisation problems are often distinct from smaller scale problems in that the various factors pertinent to the quality of a candidate solution include several which cannot easily (or at all) be expressed formally. As logistics problems get larger (hence involving greater amounts of resource and larger timescales), planners need to think increasingly in terms of political, environmental, and social factors, which are typically indeterminate. This has long been recognised; going back to one of the seminal works in algorithmic project planning [1]. E.g. a five-

year plan for a large consultancy may be affected by intervening national budgets, expected EU labour laws, the expansion plans of a nearby university, and half-expected mid-plan water shortages in the Midlands. Planners must make their best guess about how these (and many other) other factors will affect their plans, and will typically use these guesses to shape their organisation's programmes in salient ways. E.g. the planner may wish to ensure that their overall manpower demand is low during times of potential flooding, and high in September and October, but will not worry about an unduly high demand in September if he or she knows from experience that placement student availability is typically strong and likely to improve in the plan period. These and similar factors generally defy any practically useful attempts to formalise them. Even if we could imagine formalising each and every possible one of these considerations, the desirability and practicality of doing so is zero – large scale project planners simply will not be persuaded to wade through screens of difficult questions before being allowed to start building a plan.

The quantity and importance of such non-formalisable factors tasks means that such large scale problems cannot be effectively addressed by fully automated schedule optimisation systems. Indeed, such systems are very rarely used in commerce and industry. Instead, tools are used which do limited 'linear' scheduling with no attempt to optimise, but allow the user to see the consequences of any particular priority ordering among projects, and provide an interface which allows the planner to craft the overall plan by hand. With respect to the quality of a large-scale plan, what such systems can (and do) usefully do is arrange a given list of priority ordered projects in time according to given dependencies between projects, and in line with a matrix of resource availabilities (usually manpower) over time. This inevitably involves necessary shifts in the preferred start dates and durations of the various projects being scheduled, and the planner is typically able to see the extent of these 'slippages'. Beyond these easily formalised factors, however, it is up to the planner to try to arrange the projects in such a way that the slippages are acceptable (although they may be unacceptable in varying degrees to the project managers in question – these are among the unformalisable factors in the planner's expertise), *and* such that the many external issues are taken into account.

As part of a project involving a number of organisations who face such large-scale planning problems, we have developed an Interactive Evolutionary System (IES), for use by an organisation's highest level planner. Candidate plans, in the form of Gantt charts augmented by resource usage profiles, are shown on the screen together with indication of their formalisable fitness (slippage in preferred start dates and durations). The planner then supplies a score (or leaves the default score) for each plan, via a simple drop-down menu, which is designed to be an evaluation of the plan in terms of its overall quality, taking the formalisable and unformalisable factors into account. The IES uses the user-supplied scores in its assignment of selective fitness to each plan, and then produces a new population of plans for the user to assess, and the process repeats. In previous work [2], we have described the basic approach and reported on the quality of the underlying evolutionary algorithm in solving a number of large scale tasks using only internal fitness. This work also went into detail about the commercial desirability of such an IES for large scale logistics, following the results of a questionnaire survey of multi-project planners in large organisations. In summary, the basic EA we use in

the IES (which uses an indirect priority based encoding via real-number random keys [3] has been shown to outperform both Hillclimbing and Simulated Annealing on problems of realistic size, and the raw ability of the IES to achieve results *acceptable to real users* has also been proven [2]. That is, planners have been able to verify, in limited testing, that using the IES over acceptable time can lead to plans which meet their unformalised criteria with acceptable levels of slippage.

When following up that work, with the task of optimising many aspects of the IES, we were faced with an immediate concern: that of objectively evaluating versions of an optimisation system which involves intensive human collaboration, without the need to shackle experts to their PCs for months at a time. Our approach is the concept of an *Automated Tester* (AT). The AT replaces the human in the loop during experimental evaluations, and has parameters controlling its decision-making accuracy (modelling human error), and a built-in notion of a target solution which may typically be at odds with the solution which is optimal in terms of internal, formalisable fitness. The idea is that, using the AT, we can do sufficient experiments for statistical evaluation, and plausible rankings of alternative designs for the IES can be achieved. We can also allow for (and examine the effects of) different levels of human error in such studies, with the potential to design IESs which are as robust as possible to expected levels of human error. The tests so far done, and reported herein, are limited to evaluate (via the AT) the IES in two respects; whether it can be expected to work effectively on realistic problems for different kinds of target ‘shape’ which the user may drive the plan towards, and how robust the basic IES is to different levels of user error.

The remainder is set out as follows. Section 2 describes certain details of the large-scale scheduling problem for which our IES was developed, and sets out some key aspects of the problem and the approach that support understanding later sections. In section 3, we detail the flow of control in our IES, and then introduce the AT. Section 4 details experiments using the AT to determine the ability of a user to drive the plan towards specific shapes, and to evaluate robustness to different levels of human error. Section 5 is a concluding discussion.

2 Multi-Project Programme Scheduling

The application area we address here concerns the problems encountered when scheduling a large number of independent resource-constrained projects, making up an organisation’s overall programme, involving perhaps thousands of individual tasks. This is a common real-world problem, faced by many different industries (such as Manufacturing, Construction, Consultancy, R&D.), in which each project requires resources from the same finite resource pool, but needs to be scheduled as timely and efficiently as possible, based on estimated durations and preferred start and end dates specified by individual project planners. As illustrated in figure 1, an organisation’s overall programme thus consists of a number of projects, each with its own desired plan of subtasks, arranged by the manager of that project. Each project is viewed as a single ‘summary task’ which inherits the preferred start time and duration, and overall resource requirements, of the tasks within it taken

together. (lower half of the figure 1.). A single department in the organisation may have several such projects, and the Programme Manager has the job of arranging an overall plan for the entire organisation, which means organising together the summary tasks for each project in each department (upper half of figure 1.)

The nature of an organisation's overall programme varies between industries. For example, an aerospace company would have a few very large programmes, each lasting several years, made up from complex projects, each with hundreds or thousands of individual tasks. At the other end of the scale, a support department would have hundreds of individual, small projects each consisting a few tasks, with a duration of a few days or weeks at most. The individual projects are developed independently by project planners who use their knowledge and experience in defining the dates and durations as well as the layout of the project plan. The project plan represents the preferences of the project planner, who wants to minimise the potential disruption to the dates or layout of his plan, caused by the programme scheduling process. Resources tend to be scarce, and/or are required by several projects at once. These typically include people (usually defined by skills such as Lecturer, Programmer, Designer), Equipment (such as cranes, factory machines or vehicles) and Facilities (e.g. laboratories, workshops or rooms).

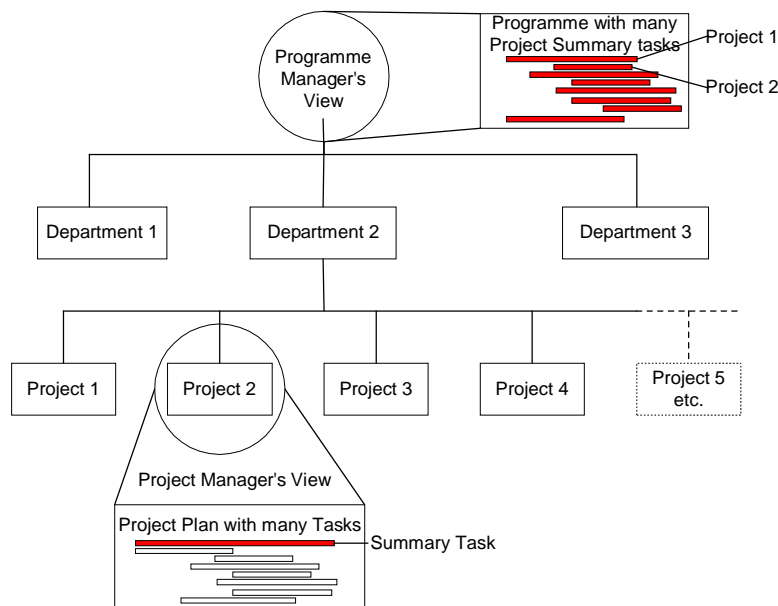


Figure 1. The structure of an organisation's overall programme.

Although evolutionary computation and other methods are well-known to work effectively on a variety of scheduling problems (e.g. [4–10]), the programme manager's concerns in the type of project in question (as explained in section 1) go

well beyond what can be formalised within a fitness function, or even a multiobjective fitness function. In this context, fully automated scheduling techniques are seldom fully acceptable to the end user, since they do not take into account this wide range of experience and intuition (or ‘gut-feel’) which the programme manager uses to judge the overall fitness of the solutions, and consequently the user often finds serious fault with the scheduler’s solutions, and does not ‘buy-into’ the process or trust the output.

However, several of the intuitive and unformalised issues which affect the plan can be cast in terms of a desired overall ‘target shape’ for the resource (usually mainly manpower) usage profile. This was hinted at in our example at the start of section 1, and is indeed a common consideration. Whether or not we can use this fact to argue for an extra formalised element to the fitness function is very arguable. That is, we could possibly construct a fully automated scheduling system which allowed the user to enter a target resource usage profile, thus compiling together many of the unformalised factors. However, this provides (undesirably) additional requirements for input, and ignores the fact that internal weighting of fit to the target profile against slippage must be determined somehow, and this, as well as the shape of the ideal profile itself, may be dependent on other aspects of the plan. We proceed in the view that formal specification of a profile would *limit* the applicability of our IES, although this remains a research issue.

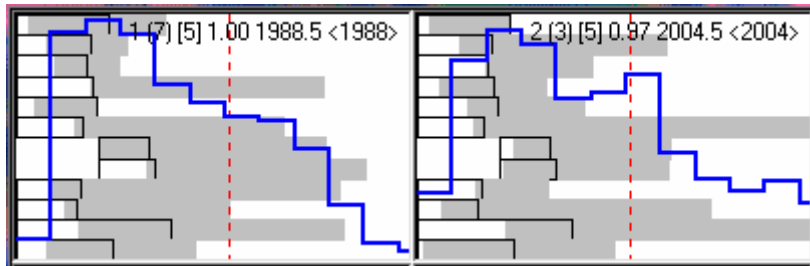


Figure 2. Two example project plans as they are shown to the user (a screen would normally contain 9 or 16 such plans in rank order).

The notion of a ‘target shape’ is illustrated in figure 2, which shows two example plans as they are viewed in our IES. Each box to the left of a row envelopes a summary task, in terms of its desired start time and duration. The grey boxes represent the placement and duration of this summary task in time in the plan itself. These are (quite typically and expectedly) far longer in duration than the project manager’s plans. The stepped line across each plan represents the resource usage profile over time (if the summary tasks all started at their planned times, this profile would need to hit the roof early on in the plan and stay there). Resource availability is the cause of plans generally stretching much further in time than planned. The series of numbers at the top of each plan summarises some key aspects, following the format **N [S] c.cc FFFF.F**, where: **N** is a rank (from 1 to size of screened population), calculated on the basis of internal fitness combined with

user supplied fitness; [S] is the Score assigned by the user (1 [Bad] - 5 [Excellent]), **c.cc** is the scaled fitness value (0.000 – 1.000) used for selection purposes, and **FFFF.F** is the internal Fitness value, in terms of days of slippage. For example, if a single project was desired to start on day 30 and last 70 days, but in a particular plan it starts on day 35 and lasts 100 days, its slippage is $(35-30)+(100-70) = 35$ days. A plan’s slippage is simply the summed slippage of its tasks. In real cases, it almost never happens that any aspect of a project’s slippage is negative.

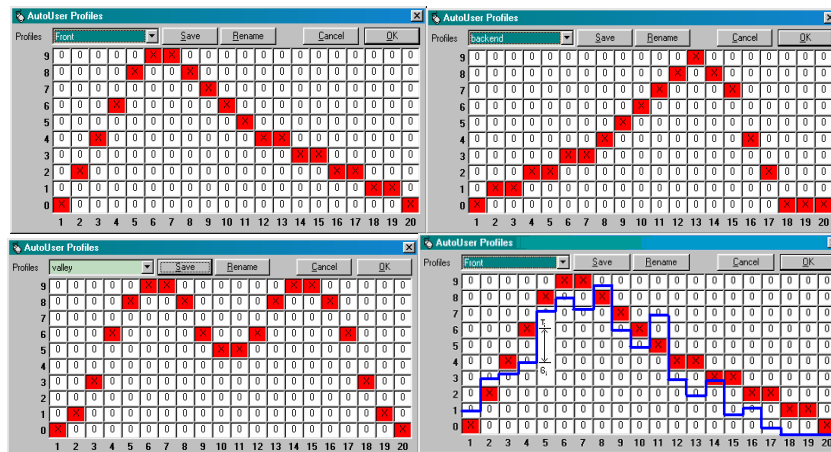


Figure 3: Target profiles used by the AT in IES evaluation experiments. Top left is a *front-loaded* profile, top-right is *back-loaded*, and bottom left is a *valley* profile. At bottom right is a desired profile aligned against an actual one.

Meanwhile, figure 3 shows the ‘shapes’, which from hereon we call profiles, that we use in later experiments as the targets for the AT driving the IES. Each of *front-loaded*, *back-loaded*, and *valley* is a typical style of desired overall resource usage profile. Another which doesn’t need illustration is a *flat* profile. At bottom right of the figure a desired profile is aligned against an actual one; the absolute differences between the two in each column are summed to yield a measure of a plan’s fit to a desired profile. In the experiments described later, the AT will typically assign Scores (from 1 to 5) to the 16 plans it sees in each generation according to the fit of each to its target profile.

3 The Interactive Evolutionary Scheduler and the Automated Tester

Figure 4 shows the flow of control in the IES. There is a growing collection of applications for evolutionary systems in which a human user essentially supplies the fitness assessment throughout the process, and our pseudocode is entirely typical of such Interactive Evolutionary Systems (e.g. see [11–19] for just a small selection). Meanwhile, an equally well-developing area includes systems where the

human interaction is sporadic, providing occasional (and necessary) guidance to the direction of an otherwise mainly automated search (e.g. [20,21,22]). This wide range of interactive EA research is testament to the hypothesis that a human user can successfully collaborate with an EA using an interactive interface, and is able to derive solutions that would have been very unlikely (i.e. very difficult and time-consuming) to have been derived by either element (human or algorithm) alone. This in turn adds further weight to the question and difficulty of objectively evaluating alternative IES designs.

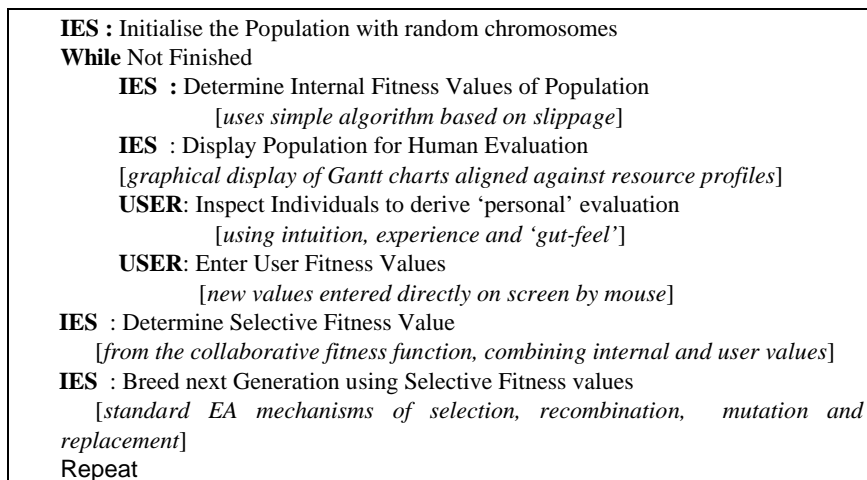


Figure 4. The Interactive Evolutionary Scheduler algorithm.

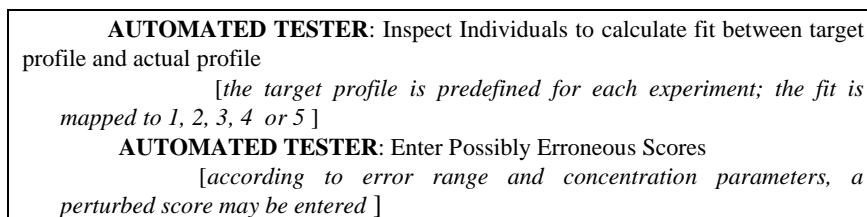


Figure 5: The IES flow of control when the AT is in use has these lines replacing the “USER” lines in Figure 4.

When the AT is in use, the two “USER” lines in Figure 4 are replaced by those in Figure 5. The AT has parameters as follows. It’s *Inaccuracy* varies between 0% and 20% representing the proportion of erroneous Scores it will assign. Its *Error range*, either 1 or 2, represents how wrong erroneous scores might be. E.g. if an error is to be made and the error range is 1, it will randomly assign a score either 1

better or 1 worse than the correct score it should assign on the basis of the target profile. It also has a *Concentration* parameter designed to make the error range and accuracy change over time in different ways; we have not yet experimented with this (awaiting research which identifies what would be plausible variations of concentration), and this is hence set to *Constant* in experiments reported here.

4 Experiments and Results

Trials of the AT looked first at the case of the ‘perfect’ user, who never makes a mistake in their relative ranking of the candidate plans on show (and also assigns those rankings against a maintained consistent target plan shape throughout the process). These tests also served to assess whether it was possible at all for reasonable target plans to be achieved (which is not guaranteed in the light of potential mismatch between the AT’s target profile and the natural shape of profiles which score well against the internal fitness measure). In all of the experiments, data for a real programme scheduling problem with 269 tasks was used, with a total Resource Usage requirement of 7252 man days. The IES had a population of 16 (as found to be suitable in preliminary experiments with human experts). In all cases, the target profile was scaled and aligned so as to define an area consistent with the total resource usage requirement of the programme being scheduled. The first set of trials were run using the ‘perfect’ user parameters in the AT; these were 0% errors and Constant Concentration. Four different target profiles were used, and 10 different runs of the AT were performed for each target. The summarised results are shown in Table 1.

Table 1. Applying the IES with the AT to a 269-task/7252 man-day programme, with varying AT internal target profiles.

Target Profile	Best Diff.	Best Percent	Average Diff.	Average Percent	Worst Diff.	Worst Percent	Std. Dev.
Front	633	8.7%	871	12.0%	1062	14.6%	117
Back	749	10.3%	903	12.4%	1074	14.8%	123
Flat	408	5.6%	605	8.3%	808	11.1%	111
Valley	858	11.8%	1033	14.2%	1345	18.5%	140

The ‘difference’ values are in units of “Man Days”. The percentage values give the difference as a percentage of the total resource usage requirement. In each case the resulting profile of the best schedule in the final population shows a surprisingly accurate fit with the target. The second set of tests was to determine how effective the algorithm could be when run with an inconsistent user, one who makes mistakes in assigning fitness to the generated solutions. These tests were done with the two more difficult profiles (those least in tune with internal fitness), which were the “Back End” and “Valley” profiles. The first tests were against the “Back End” profile with error range of 1. 10 experiments were carried out for each error level and ran for 100 iterations. The summarised results are shown in Table 2.

Table 2: Applying the IES with the AT to a 269-task/7252 man-day programme targeting a back-loaded profile, with varying AT inaccuracies and an error range of 1. In the Average column, the results for error range 2 are also given (to the right of the hyphen).

Error Rate	Best Diff.	Best %	Average Diff.	Average %	Worst Diff.	Worst %	Std. Dev.
0	749	10.3%	903	12.4%	1074	14.8%	123
1%	648	8.9%	863—883	11.9%	1160	16.0%	166
2%	565	7.8%	865—865	11.9%	1193	16.5%	186
3%	573	7.9%	843—946	11.6%	1137	15.7%	177
4%	536	7.4%	889—1076	12.3%	1208	16.7%	201
5%	699	9.6%	902—966	12.4%	1284	17.7%	176
6%	837	11.5%	970—908	13.4%	1176	16.2%	118
7%	890	12.3%	1058—1032	14.6%	1249	17.2%	127
8%	687	9.5%	1031—997	14.2%	1214	16.7%	171
10%	825	11.4%	1060—1013	14.6%	1362	18.8%	159
15%	883	12.2%	1142—1130	15.7%	1374	18.9%	173
20%	1068	14.7%	1259—1315	17.4%	1603	22.1%	174

Table 3: Applying the IES with the AT to a 269-task/7252 man-day programme targeting a valley profile, with varying AT inaccuracies.

Error Rate	Best Diff.	Best %	Average Diff.	Average %	Worst Diff.	Worst %	Std. Dev.
0	858	11.8%	1033	14.2%	1345	18.5%	123
1%	812	11.2%	1047—1070	14.4%	1294	17.8%	164
2%	783	10.8%	1100—1137	15.2%	1770	24.4%	277
3%	708	9.8%	1163—1188	16.0%	1470	20.3%	235
4%	1073	14.8%	1301—1211	17.9%	1596	22.0%	179
5%	936	12.9%	1267—1170	17.5%	1646	22.7%	280
6%	1010	13.9%	1313—1234	18.1%	1617	22.3%	197
7%	1141	15.7%	1450—1295	20.0%	1993	27.5%	279
8%	1121	15.5%	1474—1417	20.3%	1918	26.4%	227
10%	1185	16.3%	1478—1410	20.4%	1816	25.0%	234
15%	1070	14.8%	1641—1775	22.6%	2072	28.6%	346
20%	1041	14.4%	1391—1706	19.2%	1700	23.4%	254

Table 2 (and table 3) also shows the average difference in experiments with an error range of 2. For space and readability we don't give the error range 2 figures for every column. Of great interest is that the scores for the best and average differences improve compared to the perfect user for low levels of error, suggesting that low levels of user error introduce an appropriate amount of diversity which aids the search process. The second set of experiments was carried out on the same programme, with the same parameters, but this time aiming for the "Valley" profile. Table 3 summarises the results. The tests on the Valley profile also show improvements in initial results over the perfect (0% error) user for low error rates. However in this case the average difference is clearly best in the case of a perfect user, but the best differences recorded for each of the error ranges from 1% to 5%

were all better than the best recorded for the perfect user. In both cases, the IES appears surprisingly robust to an error range of 2.

5 Concluding Discussion

By replacing the human user with an AT, extensive trials can be easily done. However, before the results of such trials can be analysed, we need a way of objectively evaluating the result of a single trial. This is problematic, because the very fact that an IES is being used suggests that the quality of a solution is not readily formalisable. There are two responses to this issue: (1) We could use human expert evaluation to categorise the solutions from each trial; this is costly in terms of human expert time, however it remains a very significant saving in time compared with not using the AT at all. This is because, instead of human categorisation of a candidate solution being needed perhaps a hundred times (maybe many more) per trial, it is now needed only once per trial, to categorise the end result: (2) As we have done here, we can find some way to ‘fake’ the informal criteria by furnishing the AT with a target against which it can simply evaluate candidates, but where the conflict between that target and the formalisable criteria clearly echoes the conflict we expect between a human expert’s informal criteria and the formalisable criteria. That is, we implement a simple model of the scenario in which the user is attempting to drive the solution towards regions of the search space using criteria which are unknown to the rest of the system. and the results of a variety of experiments using the Auto-User suggested that the IGA scheduler can generally converge on a pre-defined Target profile to within about 10% of the desired ‘target’ profile, measured in man days. This has to take into account that the target profile is not necessarily feasible, and so may not be able to be exactly matched by any calculated solution.

By using the AT, we have been able to conduct some extensive experiments into the usefulness of the IES and have a number of interesting and favourable findings. First, at least given the parameters and design of the basic IES in question, low levels of human error seem to aid the process (presumably supplying extra diversity), and so such a system can be seen as robust to human error, and at (we believe in this case) levels which may be reasonable to expect from an expert user. This may not be true of a different IES configuration which perhaps has a higher mutation rate and lower pressure selection, but then again such a system may find itself *not* robust to human error, since the extra diversity will be too disruptive. Hence this finding points to a possible design principle for IESs. Similar can be said concerning the error range. The other main finding is that the IES can clearly drive a large-scale plan towards a given target profile, even a notoriously difficult one. With the IES/AT experiments this process represents a model of a user driving the plan towards acceptability to intuitive unformalised factors, and as such (and also given that details of duration and slippage were generally acceptable – details omitted for space reasons) this provides an experimental validation of the general capability and fit-for-purpose of a basic IES system for large scale programme planning.

References

1. Woodgate, H.S. (1964) *Planning by Network*, Business Publications Ltd.
2. Shackelford, M., Corne, D. (2001) Collaborative Evolutionary Multi-Project Resource Scheduling, in *Proceedings of the 2001 Congress on Evolutionary Computation*, IEEE Press, pp. 1131—1138.
3. Hartmann, S. (1999) *Project Scheduling under Limited Resources*, Springer-Verlag,
4. Tsang, E., Voudouris, C. (1995) Fast Local Search and Guided Local Search and their Application to British Telecom's Workforce Scheduling Problem, *Operations Research Letters*, **20**, pp119—127.
5. Louis, S., Xu, Z. (1996) Genetic Algorithms for Open-Shop Scheduling and Re-Scheduling, In M. E. Cohen and D. L. Hudson, editors, *Proc. of the ISCA 11th Int'l Conf. on Computers and their Applications*, pages 99–102. ISCA.
6. Bruns, R. (1997) Evolutionary Computation Applications: Scheduling, Ch 1.5, *Handbook of Evolutionary Computation*, OUP.
7. Corne, D.W., Ross, P. (1997) Practical Issues and Recent Advances in Job- and Open-Shop Scheduling, in D.Dasgupta and Z.Michalewicz (eds), *Evolutionary Algorithms in Engineering Applications*, Springer-Verlag, pp. 531—546.
8. Montana, D. (1998) Introduction to the Special Issue: Evolutionary Algorithms for Scheduling, *Evolutionary Computation*, **6**(1)
9. Bierwirth, C., Mattfield, D.C. (1999) Production Scheduling and Rescheduling with Genetic Algorithms. *Evolutionary Computation*, **7**(1), pp.1—17.
10. Husbands, P. (1999) Genetic Algorithms for Scheduling, *AISB Quarterly*, **89**
11. Dawkins, R. (1988) *The Blind Watchmaker*, Penguin.
12. Sims, K. (1991) Artificial Evolution for Computer Graphics, *Computer Graphics*, **25**(4), pp. 319—328.
13. Smith, J.R. (1991) Designing Biomorphs with an Interactive Genetic Algorithm, *Proc. 4th ICGA*, Morgan Kaufmann.
14. Graf, J., Banzhaf, W. (1995) Interactive Evolution of Images, *Proc. of Int'l Conf on Evolutionary Programming*, San Diego.
15. Aoki, K., Takagi, H. (1997) 3-D CG Lighting with an Interactive GA, *1st Int'l Conf. On Conventional and Knowledge-based Intelligent Elec. Sys. (KES '97)*.
16. Banzhaf, W. (1997) Interactive Evolution, *Handbook of Evolutionary Computing*, C2.10, OUP.
17. Johnston, V., Caldwell, C. (1997) *Tracking a Criminal Suspect through Face Space with a Genetic Algorithm*, Ch 8.3, *Handbook of Evolutionary Computation*, OUP.
18. Louis, S., Tang, R. (1999) Interactive Genetic Algorithms for the Travelling Salesman Problem, In *Proc. of the Genetic and Evolutionary Computation Conference, GECCO'99*, Morgan Kaufmann.
19. Oliver, A., Regragui, O., Monmarche, N., Venturini, G. (2002) Genetic and Interactive Optimization of Web Sites, In *Proc. Of 11th Int'l WWW Conf*, Hawaii.

20. Inoue T., Furuhashi T., Fujii M. et al., 1999, Development of Nurse Scheduling Support System using Interactive Evolutionary Algorithms. *Proceedings IEEE International Conference on Systems, Man and Cybernetics (SMC'99)*; pp 533-537
21. Parmee IC, Cvetkovic D, Watson AH, Bonham C, (2000) Multi-objective Satisfaction within an Interactive Evolutionary Design Environment. *Evolutionary Computation*, **8** (2); 197-222.
22. Parmee IC, Cvetkovic D, Bonham C, Packham I, (2001) Introducing Prototype Interactive Evolutionary Systems for Ill-defined Design Environments. *Journal of Advances in Engineering Software*, Elsevier, **32** (6);pp 429-441.