Interactive Multi-Participant Tour Allocation

Pablo Funes[1], Eric Bonabeau[1], Jérôme Hervé[2], Yves Morieux[2]
1: Icosystem Corporation, 10 Fawcett St., Cambridge, MA 02138, USA. Email: {pablo, eric}@icosystem.com
2: Boston Consulting Group, 4, rue d'Aguesseau, 75008 Paris, France. Email: {herve.jerome, morieux.yves}@bcg.com

***Abstract*- We use the example of the allocation of tours to mailmen to illustrate the general idea that Interactive Evolutionary Computation (IEC) can be applied to a range of task allocation problems where the task performers are humans. In this application of IEC, each participant is presented only with the portion of solution corresponding to his/her task (tour). In addition to the subjective evaluation of solutions by the participants, the solutions presented to the participants are pre-optimized according to objective criteria.**

I. Introduction

Many real-world optimization problems are assignment problems whereby resources are assigned a set of tasks to perform. In such problems, the objective function can be productivity, speed, robustness, etc, or a combination thereof. The optimization can be further subjected to hard or soft constraints. If the resources are human beings, there are un-verbalized and/or un-formulated and/or subjective criteria and constraints such as implicit knowledge and preferences specific to each of the participants that are not taken into account in the objective function. For example, in a factory scheduling application where workers are assigned jobs to minimize production time under such constraints as changeover times or machine compatibility, some workers may have special knowledge about the factory or may prefer to work on certain machines they like or know better; such factors are not taken into account in the objective function but may be crucial to worker efficiency, worker satisfaction and overall productivity.

In another example, mailmen are assigned tours; management's objective function is to minimize the amount of time it takes to deliver mail to increase speed of service; constraints include load balancing (each mailman should have approximately the same distance and the same load to carry) or the maximum weight a mailman can carry; certain mailmen may subjectively prefer certain tours because they know the terrain well or because the tours end close to where they live, or for other reasons unknown to the optimizer; there is also implicit knowledge a mailman has about certain tours, such as when building caretakers are present.

An evolutionary technique originally developed to generate "interesting" images and pieces of art [1-5] can be used to perform task allocation by integrating subjective criteria and subjective knowledge into the search process. The technique (see [6] for a review) is a directed search evolutionary algorithm which requires human input to evaluate the fitness of a solution (here, the fitness might be how satisfied the participants are with the tasks they have been assigned) and uses common evolutionary operators such as mutation and crossover [7] to breed the fittest solutions. Interactive evolutionary computation (IEC), as this technique is known, combines computational search with human evaluation [7]. In this paper we present an application of IEC to the general class of assignment optimization problems where the task performers are human. We use the specific example of the allocation of tours to mailmen.

In addition to the subjective evaluation of solutions by participants, the system described here also performs a pre-screening and an optimization of solutions based on objective criteria. In other words, the system combines the formalized objective function and constraints traditionally used in optimization with un-verbalized and/or un-formulated and/or subjective criteria and constraints specific to every single human participant in the system. Only the best solutions discovered in the course of the pre-optimization process are presented to the participants. The subjective evaluation of these best pre-optimized solutions by the participants is then used in a typical IEC fashion to create a new generation of solutions via evolutionary operators (selection, elitism, mutation and crossover) that is used as the starting point for another bout of automated optimization. The automated optimization process is itself population-based so as to exploit the diversity of the starting population. The whole process can therefore be viewed as a sequence of two-stage iterations, with the first stage being, for example, a fixed number of optimization steps followed by a second stage which is a single-step evaluation and fitness assignment by the participants. The result of the second stage is fed back into the first, and so forth.

Lastly, a unique feature of our approach is that each participant can only evaluate the portion of solution he/she has been assigned. In the mailmen example, that means that each mailman will only see and evaluate the tours that have been assigned to him in the various solutions, one solution being the allocation of all tours to all mailmen. Although there is a growing body of literature on interactive assignment optimization [8-13], to the best of our knowledge all examples deal with the situation of a master human planner or scheduler (or sometimes a group of experts) who interactively selects *global* schedules; one specific issue that is often dealt with is how the schedules must be visually displayed to convey the most relevant information so the user can make the best possible decisions. In our approach, it is the task performers themselves who participate in the selection of their own *local* portions of solutions. Fitness is assigned locally by the participants to the various portions of solutions and aggregated into an overall fitness for whole solutions. We believe that this constitutes a significant practical innovation as it puts the participants, with their corresponding pool of implicit knowledge and subjective preferences, back at the center of the task allocation process. This "grassroots" approach is likely to improve the participants' satisfaction while maintaining a desired level of productivity or, conversely, to increase productivity at a given level of satisfaction.

Because we did not have access to real mailmen in this work, we developed agent-based models of mailmen [14].

Each mailman is characterized by a set of properties (speed, efficiency) and a set of preferences, and ranks assigned tours accordingly. Mailman agents are described in more detail in Section II.E.

Let us emphasize that the toy example presented in this article is intended as a proof-of-concept and not as an example of a real-world implementation. Therefore a number of simplifying choices have been made. Although we run the risk of appearing simplistic, it does enable us to highlight the fundamental features of the approach:

1. The combination of automated optimization with interactive evolution.
2. The assignment of fitness values to local portions of solutions by the task performers themselves.

Applications of this approach include, but are not restricted to, crew scheduling, factory scheduling, multi-routing, service fleet planning and scheduling, task allocation, planning and scheduling in the retail, service or manufacturing industries, where it can improve worker satisfaction while maintaining or optimizing productivity, and provide means of tapping into the unexploited pool of implicit knowledge that resides with workers and employees.

II. PROBLEM, MODEL AND OPERATORS

In this section we describe our formulation of the problem and the model used to describe the spatial layout of the mail distribution zone and mailmen. Each mailman is assigned a region. Genetic operators are also described here as they are closely linked to the definition of regions and partitions.

A. Generation of Areas

We use a heuristic algorithm to create random shapes that are the borders of an arbitrary region. The algorithm works as follows: k numbers ρ_1, …, ρ_k are drawn independently from a normal distribution $N(0,1)$ of mean 0 and variance 1. The numbers are then rescaled according to: $\rho_i \leftarrow \rho_i - \min\{\rho_j\} + 1$ followed by $\rho_i \leftarrow \rho_i / \max\{\rho_j\}$, so all numbers fall between 0 and 1. The borders of the random region are the points represented by complex numbers $\rho_i \exp(2i\pi / k)$, i=1 to k.

B. Recursive cuts

We now cut the region generated as described in Section II.A, into m sub-regions using a recursive cut algorithm, where m is the number of mailmen. Let A be an area divided into s non-overlapping regions: $A = A_1 \bigcup ... \bigcup A_s$. Pick the region A_j with the largest area and divide A_j into two regions A' and A'':

$$A = A_1 \cup ... \cup A_{j-1} \cup A' \cup A'' \cup A_{j+1} \cup ... \cup A_s \quad (1)$$

This process is repeated until m regions have been created. We now have to describe how to subdivide a region into two regions: the idea is to create a random cut close to the center c of the region. Let r be the radius of the region, defined as the distance from c to the nearest point on the region's border. We first define

$$c_1 = c + r\rho \exp(2i\pi\tau) \quad (2a)$$

and

$$c_2 = c_1 \exp(2i\pi\varphi) \quad (2b)$$

where ρ, τ and φ are realizations of independent random variables drawn according to a normal distribution $N(0,1)$ of mean 0 and variance 1. We can now divide the region into two sub-regions along the line defined by the two points c_1 and c_2. ρ represents the distance from c_1 to the center with r being the distance unit, $2\pi\tau$ is the angle away from the center, and $2\pi\varphi$ is the inclination of the dividing line. Figure 1 shows a sequence of 3 recursive cuts subdividing the area into four regions

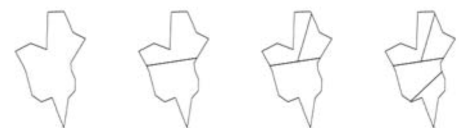


Figure 1. Example of a sequence of 3 recursive cuts.

Note that this generative procedure does not guarantee area balancing. The search engine will have to discover solutions that correspond to a fair partition of space

C. Genotype representation and mutation

Given that the number of mailmen is constant, we need a method to mutate subdivisions of the areas without altering the number of regions. This is again achieved using the recursive cut technique described in Section II.B. The area is first divided into m regions following the method described in Section II.B. The resulting genotype is a sequence of m chromosomes, each chromosome being of the form ($j_i, \rho_i, \tau_i, \varphi_i$), where j_i is the largest region, to be further subdivided, at step i of the recursive cut process, and the other parameters correspond to ρ, τ and φ in Section II.B and define a split into two sub-regions. It is important to note that the m chromosomes do not directly represent the m regions but rather represent the m successive steps of the recursive cut process. In order to mutate this particular layout of the regions, a number i between 1 and m is first selected and a uniform random noise U[-0.1,+0.1] is then applied to all three parameters $\rho_i, \tau_i, \varphi_i$. Figure 2 shows the effect of mutations on a partition.

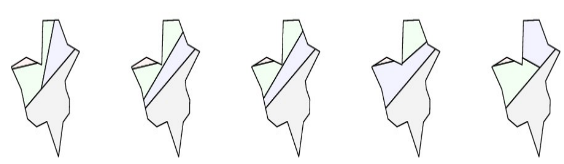


Figure 2. (1) heuristic random partition obtained from the recursive cut techniques (m=4), (2)-(5) mutations.

D. Crossover

A simple crossover mechanism is used to recombine two areas. Let us consider two partitions X and Y. X can be represented as a set of (m-1) 4-tuples $X_i=(j_i, \rho_i, \tau_i, \varphi_i)$, i=1 to m-1, where m is the number of mailmen; likewise Y can be represented as a set of m-1 4-tuples Y_i, i=1 to m-1. A uniformly random single-point crossover between two partitions X and Y is defined by a random number k drawn from a uniform distribution over $\{2,...,m-1\}$ and produces a partition Z given by: $Z_i = X_i$ if $i<k$ and $Z_i = Y_i$ if $i \geq k$. Figure 3 shows an example of a crossover. This crossover operator, which we designed for its ease of implementation, reproduces some of the properties of both parents but can certainly be improved.

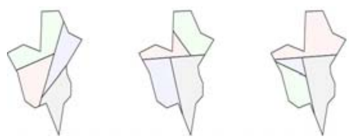


Figure 3. From left to right: partition X, partition Y and an example crossover partition Z.

E. Mailmen

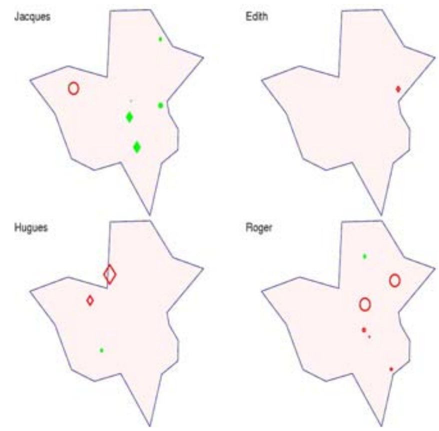
The m mailmen are represented by a set of characteristics. Some are public: they are known for example to management and can be taken into account in a traditional objective function. Some are private: they are known, sometimes only implicitly, to each mailman. Private characteristics are private because they are either not known to management or cannot be easily taken into account in a mathematical objective function.

Such characteristics as years of experience or speed of delivery are generic but we are also interested in characteristics that have geographic variations, that is, preferences and efficiencies that vary with location for a given mailman. Location-dependent variables in our model are *preferences*, which are private, and *efficiency*, which is public. These variables are assumed to be constant over most of the area to reflect the fact that most locations are probably neutral from the viewpoints of preferences and efficiency for a given mailman.

The neutral base level is assumed to be 0. In addition, certain locations are assigned a preference weight w_p and/or an efficiency weight w_e. Both w_p and w_e are drawn from a uniform distribution U[-1,+1]. For each mailman, 10 locations are randomly generated from a uniform distribution over [-1,+1] × [-1,+1]; locations that fall outside of the region are not taken into account; locations that do fall within the region are assigned a weight, either an efficiency weight or a preference weight.

A negative weight is "bad", namely the mailman is either inefficient at or unhappy with that location; a positive weight is "good", namely a mailman is either efficient at or happy with that location. At this point we assume for simplicity and demonstration purposes that mail volume is uniform across locations, but this assumption can easily be relaxed (Figure 4).

a



b

Figure 4. a: region generated. b: Preference and efficiency locations and weights for each mailman. Circles indicate preferences and diamonds indicate efficiency. Unfilled circles or diamonds correspond to negative values, filled circles or diamonds correspond to positive values. The size of the symbol is proportional to the absolute value of preference of efficiency weight.

III. Search

A. Overview

The engine works as follows: a set of partitions is generated, an optimization heuristic (here, parallel hill climbers) is run for a fixed number of iterations T to increase the *objective* efficiency of solutions; the best solutions from the optimization run are selected for presentation to the participants, who rank only the *portions of solutions* they have been assigned; selection and evolutionary operators are applied to the participant-evaluated solutions on the basis of the participants' *subjective* rankings; the resulting set of solutions are fed into the optimization heuristic, which is run again for T iterations to improve efficiency; and so forth. In this process there is therefore a sequence of two-stage iterations, the first stage being T computer-automated optimization iterations, and the second stage being one evolutionary iteration driven by a subjective fitness evaluation.

B. Efficiency Optimization

The efficiency optimization heuristic is a deterministic crowding algorithm: a population of h (here h=6) parallel hill climbers is evolved for T (here, T=200) generations. Each solution (partition) in the population is mutated, the mutated solution is evaluated and replaces its parent only if better. The population is unmixed to preserve diversity. The mutation operator is the one described in Section II.C. The efficiency score of a partition is calculated as follows. For every mailman, there is a maximum score MAX_e and a minimum score MIN_e. MAX_e is obtained by summing all positive w_e, while MIN_e is obtained by summing all negative w_e. Any given region A can then be assigned a normalized efficiency score S_e between -1 (very bad) and +1 (very good):

$$S_e = 2\left(\left(\sum_{x \in A} w_e(x)\right/\left(MAX_e - MIN_e\right)\right) - 0.5\right) \quad (3)$$

To take into account the surface area a of A, an efficiency cost function C_e is calculated:

$$C_e = a\lambda^{S_e} \quad (4)$$

where λ=0.9. According to this scoring formula, it is better (lower cost) to assign a mailman to an area where they score positive points (up to 10% improvement). This formula assumes that efficiency weights modulate the importance of the size of the area assigned to a mailman. Let $C_e(i, A)$, i=1 to m, A=1 to m, be the efficiency cost of mailman i being assigned region A. For each permutation σ which assigns a region $\sigma(i)$ to each mailman i, the total efficiency cost of σ is given by:

$$C_e(\sigma) = \sum_{i=1}^{m} C_e(i, \sigma(i)) \quad (5)$$

The best region allocation from the point of view of efficiency is

$$\hat{\sigma} = ArgMax(C_e(\sigma)). \quad (6)$$

Here, given the small size of our toy problem, $\hat{\sigma}$ is found by complete enumeration of all permutations.

Lastly, we need to take into account area balancing to ensure a fair allocation with limited size variance across the allocated regions. Regulations and limits to the number of work hours require the integration of area balancing (more generally workload balancing, since in the real world workload is affected by a multiplicity of factors such as mail volume, geographic conditions, distance walked or driven, etc). To reduce the cost variance (rather than the size variance, which does not take into account mailman efficiency)

$$\nu(\sigma) = \sum_{i=1}^{m} \left[C_e(i, \sigma(i)) - \overline{C}_\sigma\right]^2, \quad (7)$$

where $\overline{C}_\sigma$ is the average efficiency cost of a mailman for allocation σ, $C_e(\sigma)$ is modified as follows:

$$C_e(\sigma) \leftarrow C_e(\sigma)\psi^{\nu(\sigma)}, \quad (8)$$

where ψ =1.3. In other words, $C_e(\sigma)$ is amplified if region allocation σ is uneven. C_e is the cost function used to evaluate solutions in the parallel hill climbers. Better solutions have lower C_e values.

C. Interactive Evolution

The n (here, n=6) best solutions from the optimization run with the objective function C_e are now presented to the mailmen, who assign a dissatisfaction number to the different solutions based solely on the regions they have been assigned (they either have no knowledge of which regions the other mailmen are assigned, or do make use of that knowledge in rating their own regions). Each mailman is by definition assigned one region in each of the six partitions. A solution is a partition P= $A_1, \ldots, A_m$ of space into m regions AND an assignment σ of mailmen to the regions. The process is similar to the one defined in Section III.B. For every mailman, there is a maximum preference score MAX_p and a minimum preference score MIN_p. MAX_p is obtained by summing all positive preference weights w_p, while MIN_p is obtained by summing all negative w_p. Any given region A can then be assigned a normalized preference score S_p between -1 (very bad) and +1 (very good):

$$S_p = 2\left(\left(\sum_{x \in A} w_p(x)\right/\left(MAX_p - MIN_p\right)\right) - 0.5\right) \quad (9)$$

To take into account the surface area a of A, a preference cost function C_p (or dissatisfaction) is calculated:

$$C_p = a\lambda^{S_p} \quad (10)$$

where λ=0.9. Each mailman can therefore compute a personal dissatisfaction for each of the n regions they have been proposed. Based on C_p, each mailman can now rank each of the n solutions. Let $R(i,k)$ be mailman i's ranking of solution k. The cost $C_p(k)$ of solution k is the sum of the cubed rankings:

$$C_p(k) = \sum_{i=1}^{m} R^3(i, k) \quad (11)$$

The cubic exponent is aimed at punishing solutions that give any mailman a high preference cost, thereby implicitly favoring preference balancing.

Based on the calculation of C_p, a new population of n=6 solutions is created as follows:

(1) the best solution remains unchanged;

(2) a mating pool comprised of the best three solutions is built;

(3) two new solutions are created by applying the crossover operator described in Section II.D to randomly selected partitions in the mating pool and finding the best permutation (lowest dissatisfaction);

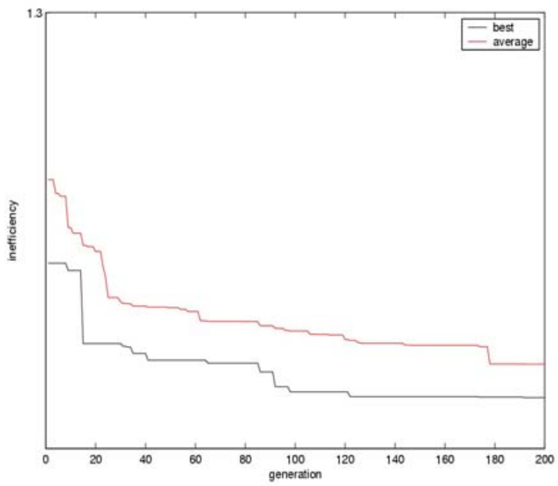
(4) two new solutions are created by applying the mutation operator described in Section II.C to randomly selected partitions in the mating pool and finding the best permutation (lowest dissatisfaction);

(5) one new solution is generated randomly.

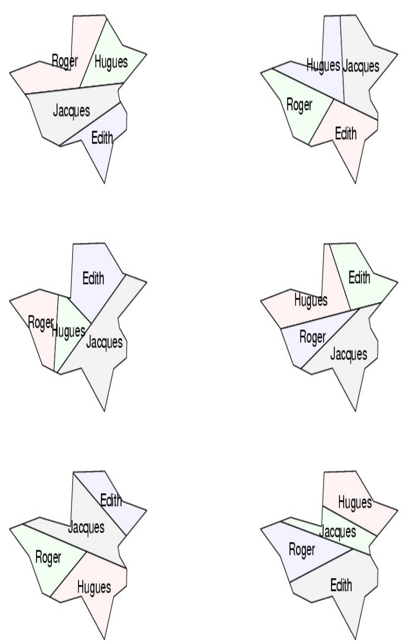
IV. Results

The results presented in this section have been obtained with four mailmen. First, a region is generated following the procedure described in Sections II.A and II.B. Positive and negative preference and efficiency locations are then assigned to each mailman. The efficiency optimization heuristic is then run for T=200 iterations.

Fig. 5a shows a typical (average and best) fitness plot during the optimization run; here, fitness is in fact a cost function that needs to be minimized (C_e). Fig. 5b shows the six best partitions from the optimization together with their best corresponding allocations.



a



b

Figure 5. Efficiency optimization run (a) and six best solutions obtained after 200 generations with 6 parallel hill climbers (b). A solution is the combination of a partition and an allocation of mailmen.

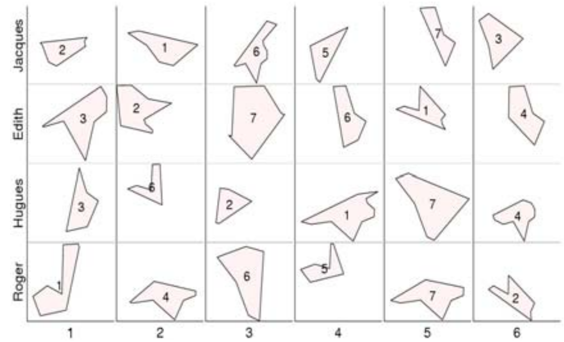
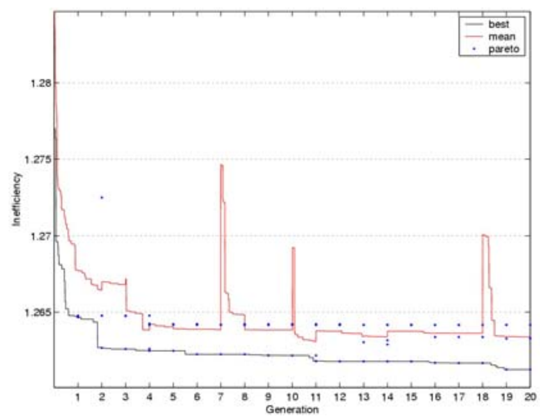
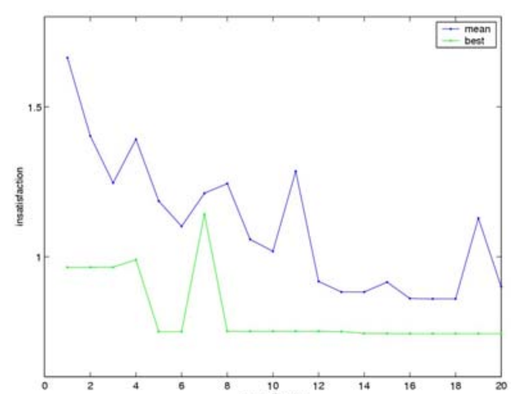


Figure 6. Solutions presented to the mailmen and ranks. Note that each mailman only ranks the regions he or she has been assigned.

The six best solutions are then presented to the mailmen in the form of the regions they have been assigned. Each mailman ranks each region. Figure 6 shows how solutions are presented to the mailmen and each mailman's ranking.



a



b

Figure 7. (a): efficiency fitness plot (average and best) over 20 iterations of the process. Note that the efficiency cost function often increases after the population has been modified via the preferences of the mailmen. For each generation, points on the inefficiency-dissatisfaction Pareto frontier are also shown. (b): dissatisfaction fitness plot (mean and best).

A new population is generated based on the mailmen's rankings. This process is iterated until satisfactory solutions are discovered. Figure 7 shows fitness plots for efficiency and preferences (both fitness functions are in fact cost functions that need to be minimized, C_e for inefficiency and C_p for dissatisfaction) for 20 iterations of the two-stage process (efficiency optimization [200 iterations] + preference evaluation [1 iteration]). Both plots show a significant decreasing trend. A decreasing trend, however, is not sufficient: the problem is a multi-objective optimization problem where one seeks to maximize efficiency and minimize dissatisfaction, and looking at single-objective plots does not tell us whether we are making progress on both fronts. Indeed the best efficiency solution may be a bad preference solution and vice versa. While there is an extensive literature on evolutionary algorithms for multi-objective optimization (see for example, Fonseca & Fleming, 1995), the present setup is unique in that one objective is explicit while the other one can only be assessed by the participants.

A Pareto plot is necessary to evaluate how the process is doing on both objectives simultaneously. Figure 8 shows inefficiency vs dissatisfaction at iterations 1, 2, 4, 8, 16 and 20. The best solutions, which minimize both efficiency cost and preference cost, are to be found in the lower left corner of the plot. It is clear that the best multi-objective solutions are significant improvements over the ones of iterations 1 and 2. The Pareto fronts are clearly moving into the right region.

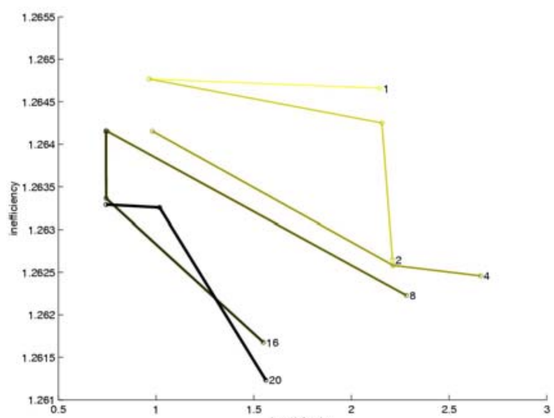


Figure 8. Dissatisfaction vs inefficiency Pareto plot at the end of iterations 1, 2, 4, 8, 16 and 20.

V. Discussion

We have illustrated with a simple model of tour allocation (in fact an even simpler form of tour allocation, zone allocation) in mailmen how IEC can applied to task allocation problems with subjective *participant* criteria. In this approach, IEC is multi-participant in the sense that several participants evaluate solutions and assign fitness; in addition, each participant only evaluates the portions of solutions that correspond to their proposed assignments, which is radical departure from typical multi-user IEC where every participant has access to and can evaluate entire solutions. This approach has the advantage of reducing the amount of information necessary to perform evaluations and minimizes decision-making interferences between participants.

The pre-optimization of solutions before they are presented to the participants also reduces participant workload, which is very important in IEC. One question, however, was whether traditional optimization and IEC could collaborate and move their respective objectives in the right direction simultaneously. The example described in this paper is obviously a toy model but one that does show that the approach works: the Pareto fronts for the combined objectives make progress toward solutions that improve both objectives.

We assumed no correlations between (1) efficiency and preferences, and (2) between mailmen in the definitions of personal efficiencies and preferences, which makes the problem neutral in two ways: (1) positive correlations between efficiency and preferences would have enhanced the symbiosis between both components of the system, but negative correlations would have made it harder for the two components to work synergistically; (2) the degree of overlap between mailmen preferences and efficiencies influences the difficulty of finding solutions that satisfy everyone; The approach, however, constitutes the first step toward integrating traditional optimization with subjective criteria and implicit knowledge by the collective of task performers and as such has a wide range of possible applications.

References

[1] Dawkins. R. 1987. *The Blind Watchmaker*. W. W. Norton, New York.

[2] Sims, K. 1991. Artificial evolution for computer graphics. *Computer Graphics* **25**: 319-328.

[3] Sims, K. 1992. Interactive evolution of dynamical systems. Pages 171-178 in: *Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life* (F. J. Varela & P. Bourgine, eds.), MIT Press, Cambridge, MA.

[4] Sims, K. 1993. Interactive evolution of equations for procedural models. *Vis. Comput*. **9**: 446-476.

[5] Banzhaf, W. (1997) Interactive evoution. In: *Handbook of Evolutionary Computation* (T. Baeck, D. Fogel, and Z. Michalewicz, eds), Ch. C2.10, pp. 1-5, Oxford University Press.

[6] Takagi, H. 2001. Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation. *Proc. IEEE* **89**: 1275-1296.

[7] Michalewicz, Z. 1999. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer.

[8] Inoue, T., Furuhashi, T., Fujii, M. *et al*. 1999. Development of nurse scheduling support system using interactive EA. In: Proc. IEEE Int. Conf. on Systems, Man and Cybernetics (SMC'99), vol 5, pp. 533-537, IEEE Press.

[9] Klau, G.W., Lesh, N.B., Marks, J.W., Mitzenmacher, M. 2002. Human-guided Tabu Search. In: *National Conference*

on Artificial Intelligence, ISBN: 0-262-51129-0, pp. 41-47, AAAI Press.

[10] Lesh, N., Lopes, L. B., Marks, J., Mitzenmacher, M., Schafer, G. T. 2003. Human-guided search for jobshop scheduling. In: *3rd International NASA Workshop on Planning and Scheduling for Space. Houson, Texas.*

[11] Scott, S. D., Lesh, N., Klau, G. W. 2002. Investigating human-computer optimization. In: *Proc. ACM Conference on Human Factors in Computing Systems (CHI)*, pp. 155-162, ACM Press.

[12] Anderson, D., Anderson, E., Lesh, N.B., Marks, J. W., Perlin, K., Ratajczack, D., Ryall, K. 1999. Human-guided simple search: combining information visualization and optimization. In: *Proc. ACM Workshop on New Paradigms in Information Visualization and Manipulation (NPIVM)*, pp. 21-25, ACM Press.

[13] Bracklow, J. W., Graham, W. W., Hassler, S. M., Peck, K. E., Powell, W. B. 1992. Interactive optimization improves service and performance for Yellow Freight system. *INTERFACES* **22**, 147-172.

[14] Bonabeau, E. 2002. Agent-based modeling: methods and techniques for simulating human systems, *Proc. Nat. Acad. Sci. USA* **99**, 7280-7287.

[15] Fonseca, C. M., Fleming, P. J. 1995. An overview of evolutionary algorithms in multi-objective optimization. *Evolutionary Computation* **3**, 1-16.