

# Map-labelling with a Multi-objective Evolutionary Algorithm

Lucas Bradstreet  
School of Computer Science &  
Software Engineering  
The University of Western  
Australia  
35 Stirling Highway  
Crawley, Australia, 6009  
lucas@csse.uwa.edu.au

Luigi Barone  
School of Computer Science &  
Software Engineering  
The University of Western  
Australia  
35 Stirling Highway  
Crawley, Australia, 6009  
luigi@csse.uwa.edu.au

Lyndon While  
School of Computer Science &  
Software Engineering  
The University of Western  
Australia  
35 Stirling Highway  
Crawley, Australia, 6009  
lyndon@csse.uwa.edu.au

## ABSTRACT

We present a multi-objective evolutionary algorithm approach to the map-labelling problem. Map-labelling involves placing labels for sites onto a map such that the result is easy to read and usable for navigation. However, map-users vary in their priorities and capabilities: for example, sight-impaired users need to maximise font-size, whereas other users may be willing to accept smaller labels in exchange for increased clarity of bindings of labels to sites. With a multi-objective approach, we evolve a range of labellings from which users can select according to their particular circumstances. We present results from labelling two maps, including a difficult, dense map of Newcastle County in Delaware, which clearly illustrate the advantages of the multi-objective approach.

**Categories and Subject Descriptors:** I.2.1 [Artificial Intelligence]: Applications and Expert Systems — cartography, G.1.6 [Numerical Analysis]: Optimization — constrained optimization

**General Terms:** algorithms, experimentation

**Keywords:** evolutionary algorithms, multi-objective optimisation, map-labelling.

## 1. INTRODUCTION

The task of *map-labelling* involves taking a map marked with a number of *sites*, each of which has a name or *label*, and arranging these labels on the map such that the result is clear enough for the map to be useful for a given user and task. In this definition, “clear enough” means that the labels are readable by the user, and that it is easy to identify the label that is associated with each site and vice versa. Map-labelling is both an important task (maps are used in a wide range of applications) and a difficult one (the number of labellings for a given map is essentially infinite). Since users vary widely in their abilities and requirements, different la-

bellings will suit different users. As such, map-labelling is usually done by hand. Automatic map-labelling offers clear advantages, both in terms of the speed and cost of labelling, and in terms of adapting labellings to the needs of particular users. In addition, online map generation requires maps to be generated automatically. Such maps clearly cannot be labelled by hand.

The goal of generating clear labellings can be broken down into a collection of sub-goals [9], and as such, map-labelling is a *multi-objective* problem. The essence of multi-objective problems is that candidate solutions are judged according to multiple criteria that usually conflict with each other to some degree. Thus, a good solution can be improved in one criterion only by accepting worse performance in at least one other criterion. The goal with multi-objective problems is to generate a set of solutions that compromise in different degrees between the different criteria — the solution to be used in a given situation can be selected according to the particular needs of that situation, for example, the abilities of a given user, or prevailing market conditions, or expected weather conditions, or so on. This requirement to generate a set of solutions for a multi-objective problem fits well with the basic model of evolutionary algorithms, where a population of solutions is evolved under the influence of a fitness function. Multi-objective evolutionary algorithms (MOEAs) have been applied successfully to many problems in the past [4, 1, 3, 13].

We present a MOEA approach to map-labelling that evaluates solutions according to three objectives:

1. *font-size* — within reason, bigger labels implies that a map is easier to read;
2. *conflicts* — labels should not overlap with each other, nor with sites;
3. *clarity* — it should be easy to tell which label belongs to a given site, and vice versa.

This approach generates a range of maps that compromise to different degrees between the various objectives. For example:

- it finds automatically the biggest font-size that can give zero conflicts and maximises clarity at that font-size;
- it allows a user to improve map clarity if they can accept a smaller font-size;

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '05, June 25–29, 2005, Washington, DC, USA.  
Copyright 2005 ACM 1-59593-010-8/05/0006 ...\$5.00.

- it allows a user to accept a partially-labelled map if they need greater clarity or a bigger font-size; and
- it allows a range of other compromises to suit the needs of different users.

Thus, different users can select different labellings, depending on their individual needs.

The rest of this paper is structured as follows. Section 2 describes previous approaches to the problem of map-labelling. Section 3 describes the essential features of our multi-objective approach. Section 4 gives results for two example maps: Southern Guam, with 41 sites, and Newcastle County in Delaware, with 185 sites. Section 5 concludes the paper.

## 2. PREVIOUS APPROACHES

A seminal paper by Imhof [9] outlined several requirements for good map labelling. These have formed the basis for all further work in the area. These requirements are:

- *Legible* — labels must have legible font-sizes and be positioned in such a way that they are easily read.
- *Unambiguous* — each label must clearly identify a single site and not be confused with another label or site.
- *Overlap avoidance* — a label should not overlap any other label or site.
- *Aesthetics* — labels should not be overly clustered or distract from map features.

Cartographers attempting to solve the map labelling problem must attempt to satisfy these subjective quality criteria in order to achieve high quality solutions. All attempts at automated map labellers include some subset of these requirements to form their optimisation criteria.

Previous approaches tackle this optimisation process using a variety of methods. Raidl [12] uses a Genetic Algorithm (GA) combined with heuristic techniques to avoid conflicts. Raidl finds that this gives better results than another approach by Christensen et al. [5] that uses simulated annealing (SA) as its optimisation technique. A variation on the Raidl's GA was proposed by van Dijk et al. [15], which applies local optimisers in the place of a mutation operator.

Each of the above approaches limits labels to eight possible positions around a site and define a fitness function calculated using the number of overlapping labels and the desirability of different label positions. Under this representation, the fitness of a map-labelling  $M$  is calculated as:

$$f(M) = \sum_{i=1}^n (c_i + \frac{(r_i - 1)}{N}), c_i \in \{0, 1\}, r_i \in \{1..N\}$$

where  $n$  is the number of sites for  $M$ ,  $N$  is the number of possible label positions,  $c_i$  is a binary value equalling one when a conflict occurs for label  $i$  and zero otherwise, and  $r_i$  is the preference value of the current label position for label  $i$ . These approaches primarily differ in the optimisation strategy used rather than the function of labelling quality being minimised.

Preuß [11] demonstrates an evolutionary strategy (ES) for solving the map labelling problem using a polar coordinate scheme to represent label positions, allowing many more label positions to be taken than the previous approaches outlined above. Under Preuß's ES, the fitness function incorporates more than label to label conflicts when evaluating the

fitness of a candidate solution. The fitness function factors in a variety of label overlap conflicts, a label to site distance measure, and position preferences. The fitness function includes a series of weights that are chosen based on the relative importance of each factor.

While SA and GA map labellers have proved useful when used to minimise conflicts, they have not been used to optimise for other objectives that relate to the overall quality of a labelled map. Even though conflicts are an important measure of map quality, Imhof [9] discusses a number of competing considerations for producing maps of high quality. For example, neither the SA or GAs described previously attempt to optimise for the font-size or clarity of map labels, each of which has an effect on the overall usability of a map.

## 3. A MULTI-OBJECTIVE APPROACH

Following the work outlined in Section 2, we restrict our experimental investigations to problems containing only point sites and do not include more complex sites such as lines (for example rivers) or areas (for example regions) in our maps. We also restrict the problem to constrain all labels within a particular labelling to use the same font-size.

The principal issues in the application of a MOEA to a problem are the definitions of and interactions between the objectives, the choice of representation, and decisions about the various parameters involved in the execution of the algorithm. We omit the aesthetics objective because it is difficult to quantify.

### 3.1 Definition of objectives

We use three objectives that represent different priorities that users might place on a labelling.

**Font-size** Within reason, we'd like to use the largest font-size that fits on the map, as this is likely to make the map easier to read quickly. However, a larger font-size is likely to involve compromise on the other two objectives. In order to cast the problem uniformly in terms of minimisation, we define  $FS$  to be the difference between a pre-determined largest font-size and the given font-size. So, in order to maximise font-size, we need to minimise  $FS$ .

**Conflicts** Labels should not overlap with each other, nor with the sites on the map. Following the ideas of Preuß [11], we weight different types of conflicts according to their perceived importance: label/label conflicts are weighted at 2, label/site conflicts are weighted at 3, and label/map-edge conflicts are weighted at 8. The weights of all the conflicts on the map are summed and the result minimised. Where conflicts cannot be avoided involving a given subset of the labels, a choice has to be made between omitting some of those labels, or living with the overlaps.

**Clarity** It should be clear which label belongs to each site, and vice versa, ideally with a single glance at the map. Labellings may be unclear in the sense that they have local ambiguities which can be resolved by taking a more-global perspective (e.g.  $L1$  and  $L2$  are equidistant from  $S1$ , but elsewhere it is clear that  $L2$  belongs to  $S2$ , so  $L1$  must belong to  $S1$ ). Worse are labellings which have ambiguities that cannot be resolved at all,

in which case the map is probably unusable. These differences are illustrated in Figure 1.

We define a measure of clarity called *ROD* based on the ratio of the distance to the site’s own label and the distances to other sites’ labels. The *ROD* of a map-labelling  $M$  is calculated as:

$$ROD(M) = \sum_{s \in S} \sum_{s' \in S} \frac{dist(l_s, s)^2}{dist(l_{s'}, s)^2}$$

where  $S$  is the set of sites,  $l_x$  is the label of site  $x$ , and  $dist(y, x)$  gives the Euclidean distance between the site  $x$  and the closest point on label  $y$ . Map clarity is maximised when the *ROD* is minimised.

Using the definitions above, the map-labelling problem is defined as a multi-objective optimisation problem involving three objectives (*FS*, conflicts, and *ROD*) that need to be minimised. Note that the relative importance of these objectives varies from user to user and from map to map: this is the fundamental reason for taking a multi-objective approach to the problem.

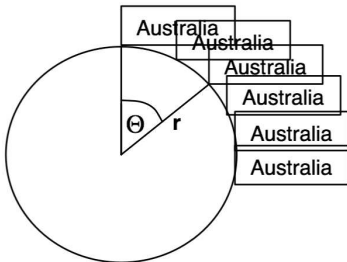
### 3.2 Representation of labellings

There are two components in the representation of a labelling. First, each individual has an integer that encodes the *FS* used in the labelling. The major component of the representation is the set of label positions, one for each site on the map. Label positions are represented using continuous distance and angle co-ordinates, allowing the label for a site to be placed anywhere around that site. However, using simple polar co-ordinates to a fixed point on the label gives continuity problems: often small changes in the values in the representation give large changes in label position, and vice versa. We use a *sliding label scheme* due to Hirsch [8] and extended by Preuß [11], where each label position is represented by a pair  $r, \theta$ :

$r$  is the distance between the site and the closest point on the bounding box of its label, and

$\theta$  is the angle from the site to the closest point on the bounding box of its label, modified slightly to account for cases where the label crosses the horizontal or vertical lines from the site.

This scheme is illustrated in Figure 2. We also define a



**Figure 2: Demonstration of various label positions under the sliding label scheme.**

minimum value for  $r$  to prevent labels conflicting with their own sites, and a maximum value, to disallow positions that are inherently unclear.

Thus for a map with  $n$  sites, the representation of an individual contains  $2n + 1$  values which are optimised by the evolutionary algorithm.

### 3.3 Algorithm details

To perform the optimisation, we use a fairly straightforward evolutionary strategy with the following parameters:

- Population size: 400.
- Mutation probability:  $2/n$  for labels; 0.15 for font-size.
- Recombination: uniform crossover with probability 1.0; 4 parents.
- Selection: non-dominated sort with crowdedness comparison-operator by Deb et al. [6].
- Local optimisation: applied every 80 generations, and additionally at the end of the last generation.

Details of experiments to determine these values can be found in Bradstreet’s thesis [2].

Selection uses Goldberg’s Pareto-ranking of individuals [7] and a crowdedness comparison-operator by Deb et al. [6] to discriminate within ranks where necessary. However, we enhance the standard definition of dominance to accord somewhat more significance to the conflicts objective than to the other two objectives:

$$\begin{aligned} \text{dominates}(X, Y) \quad \text{iff} \quad & \text{paretoDominates}(X, Y) \text{ or} \\ & \text{conflicts}(X) \leq \text{conflicts}(Y) - 10 \\ & \text{and } FS(X) \leq FS(Y) \end{aligned}$$

where *paretoDominates*( $X, Y$ ) is defined to return *true* if and only if  $X$  is “better” than  $Y$  in at least one objective and at least equal to  $Y$  in all other objectives.

This modified definition for *dominates* reflects the fact that maps with too many conflicts are inherently difficult to read and are unlikely to be acceptable. Note however that a dominated labelling is not necessarily condemned; it is just less likely to survive and reproduce.

In addition, we apply local optimisation at various stages of the algorithm to better improve overall performance, resulting in a form of memetic algorithm [10]. Local optimisation consists of a “hill-climbing” local optimiser which makes small changes to a label’s position, accepting the new position only if it dominates the old one. Labels are processed in a randomised order to promote diversity, as in Raidl [12]. A similar technique was used in the map-labelling GA of van Dijk et al. [15] in lieu of mutation.

## 4. EXPERIMENTAL RESULTS

To show the effectiveness of our MOEA map-labelling system, we report results on a series of experiments performed on two maps of varying difficulty:

1. **Southern Guam** — a relatively small map with 41 sites. Figure 6 plots three potential map-labellings for this map. Two regions are difficult to label — the first midway down the right-hand side where a number of sites (Asagas, Mayaya, Asfaja, Tumon Chama, Peca, Chagamin Lago, and other surrounding sites) are tightly clustered; the second in the bottom left corner where five sites (Talona, Toa, Joatan, Lingae, and Julog) are close together.

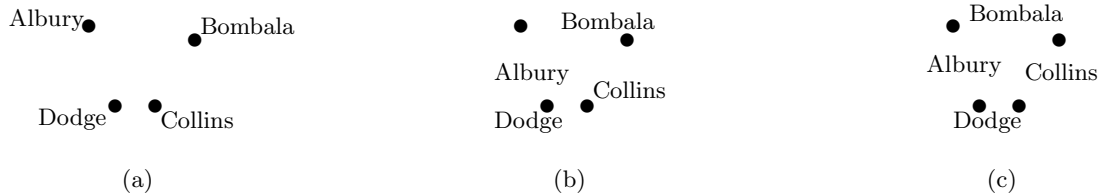


Figure 1: Three labellings of varying clarity. (a) is an ideal labelling, (b) is somewhat less clear, (c) is probably unusable.

2. **Newcastle County, Delaware** — a harder map with 185 sites. The combination of long label names and the density of sites causes difficulties, especially at larger font-sizes. Figure 7 plots two potential map-labellings for this map.

Data for site coordinates are translated from longitude and latitude values taken from GNIS databases [14].

Each label is represented by the single bounding-box that contains the entire text of the label. We choose an upper-case font for label names to ensure better visual correspondence between the displayed position of the label relative to its site and the algorithm’s “perception” of its proximity<sup>1</sup>. Conflicts are registered when a label bounding-box either intersects one or more of the borders of the map, intersects another label bounding-box, or contains a site within its bounding-box. As explained in Section 3.1, conflicts are counted in terms of the number of labels involved, adjusted for the severity of the type of conflict.

#### 4.1 Experiment 1

In this experiment, we report the results of five different runs of MOEA map-labelling system for the map of Southern Guam, each run lasting 500 generations. The MOEA attempts to optimise for seven font-sizes: 12pt to 18pt inclusive.

Figure 3 plots the final non-dominated Pareto fronts for the smallest and largest font-sizes for these five runs.

We observe from Figure 3 that for the 12pt font-size, each of the five runs has converged to approximately the same non-dominated Pareto front. Tight convergence to a similar non-dominated Pareto front suggests that the problem is relatively simple at this font-size as the MOEA is repeatedly able to find (high quality) solutions of about the same quality. We also observe that the MOEA has found a zero-conflict map-labelling for each run for the 12pt font-size, thus suggesting this approach is applicable to generation of zero-conflict maps in a relatively short amount of time.

Alternatively, for the 18pt font-size, Figure 3 shows far less convergence to the same non-dominated Pareto front for the five different runs. In this case, we see more variance in final Pareto fronts, especially at low numbers of conflicts. This suggests that the problem is harder at this font-size than at the smaller font-size. This is as expected — as the font-size increases, labels take up more space in the map and hence are more likely to overlap with other labels. Added to the stochastic nature of the MOEA, the increased diffi-

<sup>1</sup>Better visual correspondence can be achieved by using bounding-boxes for each character in the label, however overall running time is greatly increased.

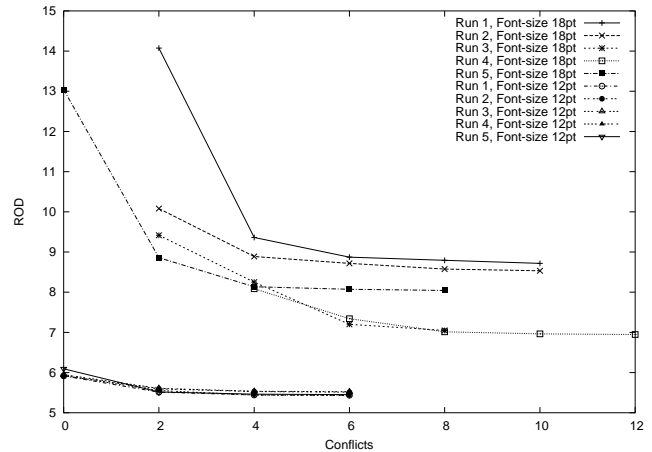


Figure 3: Final non-dominated Pareto fronts for the smallest and largest font-sizes for five runs of the MOEA map-labelling system for the map of Guam.

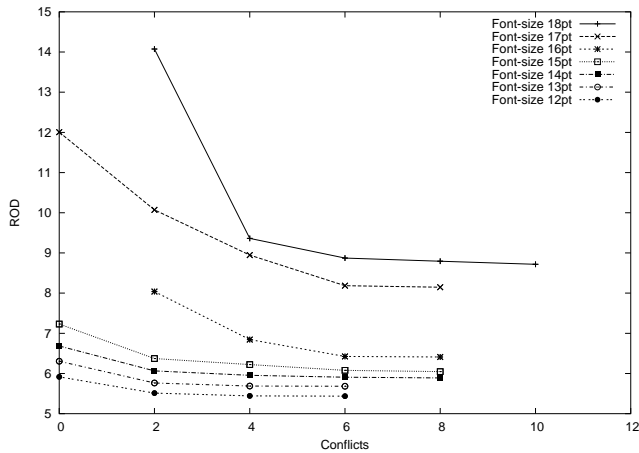
culty in arranging the labels at this font-size without conflict suggests that more time is needed to ensure convergence to good solutions for this font-size.

Figure 3 also shows that the MOEA was able to find a zero-conflict map-labelling for the 18pt font-size in only one of the five runs (Run 2). While three of the other four runs were able to find a two-conflict map-labelling, one run (Run 4), was unable to find a two-conflict map-labelling. As expected, we observe that the measure of clarity for the 18pt zero-conflict map-labelling is relatively poor, although oddly it is better than the measure of clarity obtained for the 18pt two-conflict map-labelling of Run 1. Again, this highlights the difficulty of this problem at this font-size.

Returning to the 12pt font-size, Figure 3 shows that all five runs produce a final non-dominated Pareto front containing no more than six conflicts at this font-size. It seems no better clarity can be obtained by allowing more than six conflicts; indeed, all 7-conflict and higher map-labellings are dominated by other members in the final population. The relatively small font-size means labels can be arranged in such a way that only six conflicts are needed to maximise overall map clarity.

Figure 4 plots all the final non-dominated Pareto fronts for the different font-sizes of Run 1 from Figure 3.

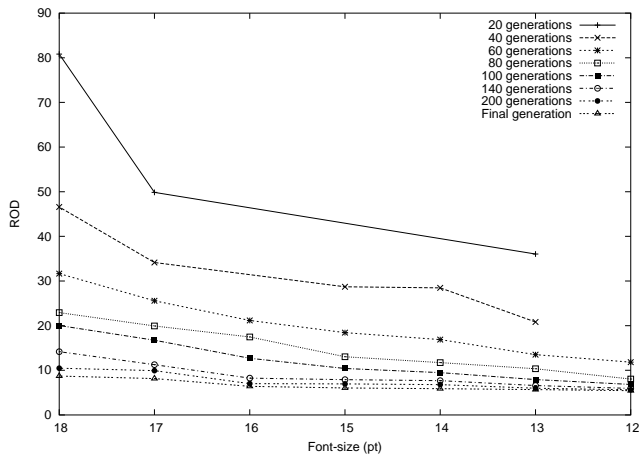
We observe from Figure 4 that the MOEA is able to find zero conflict map-labellings for five of the seven font-sizes for this run. Convergence to similar quality solutions is ob-



**Figure 4: Final non-dominated Pareto fronts for the different font-sizes of Run 1 from Figure 3 for the map of Guam.**

served at the four lowest font-sizes, however more variability in the quality of solutions at the the larger font-sizes is evident. As before, this is as expected — increased font-size necessarily makes the problem more difficult as labels occupy more space in the map.

Figure 5 shows the evolution of Run 1 from Figure 3 through time. For each of the reported generations, the non-dominated Pareto front considering only the clarity measure and font-size objectives is plotted. As expected, we observe a general convergence towards the origin of the graph (recall that the objectives are being minimised) over the course of the run. We also observe asymptotic improvement in solution quality typical of optimisation algorithms — relatively large gains are obtained in the first few reported generations compared to the later generations.



**Figure 5: Progress through generations for Run 1 from Figure 3 for the map of Guam.**

Similar plots comparing the other combinations of the objectives (*ROD* versus conflicts and conflicts versus font-size) show a similar trend, but for brevity, have been omitted.

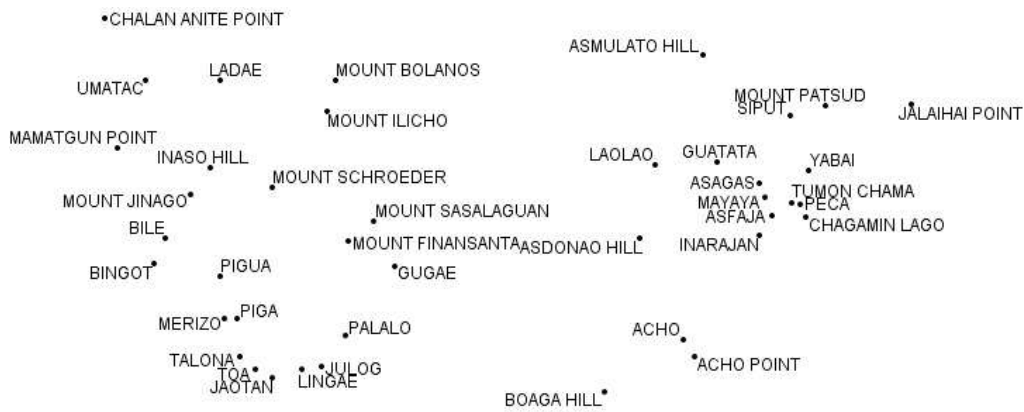
Figure 6 plots three non-dominated map-labellings for

Run 1 from Figure 3. Figure 6(b) plots the final non-dominated map-labelling with the largest font-size that contains no conflicts. This map contains no conflicts and has a relatively large font-size, but has poor (relative to the other non-dominated map-labellings) clarity. Most sites on the map are labelled well (there is no ambiguity about which site each label is attached to); the relatively poor clarity can be attributed to a few sites: Lingae (in the bottom left corner), Tumon Chama, and Peca (both midway down the right-hand side) contribute most to the relatively poor clarity of this map. Due to the large font-size and the need for zero conflicts, these labels are “pushed” further away from their sites than desired, creating ambiguity about which sites these labels are actually attached to. Indeed, examining Figure 6(b), shows that some deductive work is required to correctly determine which site the Lingae label refers too. Only after attaching the surrounding labels to their most obvious sites can we determine the correct site for the Lingae label. To a lesser extent, this is also true for the ambiguity created for the Tumon Chama and Peca labels midway down the right-hand side of the map.

The clarity of Figure 6(b) can be improved by either decreasing the font-size, increasing the number of conflicts or both. Figure 6(a) plots the final non-dominated map-labelling that contains the same number of conflicts as Figure 6(b), but with a font-size of 12pt instead of 17pt. Correspondingly, the map-labelling represented in Figure 6(a) has a *ROD* value of 5.92 compared to a *ROD* value of 12.01 for the map-labelling shown in Figure 6(b). Examination of Figure 6(a) shows this decrease in *ROD* value corresponds to a better clarity map — the ambiguity of which sites the Lingae, Tumon Chama, and Peca labels are attached to is greatly reduced. Indeed, there is very little ambiguity about the true attachment of any label in this map-labelling. However, the font-size for Figure 6(a) is significantly smaller than Figure 6(b). This decreases readability, especially for sight-impaired readers.

Another way of increasing the clarity in Figure 6(b) is to compromise on the number of conflicts and allow intersections between two or more labels, or labels and sites. Figure 6(c) plots the final non-dominated map-labelling with the same font-size as Figure 6(b), but with four conflicts instead of none. Comparison of Figure 6(c) to Figure 6(b) shows that the clarity of this map-labelling is also increased — again, there is very little ambiguity about which sites the Lingae, Tumon Chama, and Peca labels are attached too. However, the cost for this improvement in clarity comes in the increase in conflicts. In this map-labelling, we see that the Lingae label intersects with the neighbouring Joatan label and that the Tumon Chama and Peca labels intersect with each other. After some consideration, this is as expected — these labels were the cause of the poor clarity measure of Figure 6(b), hence significant improvements in clarity can be made by moving these labels closer to their correct sites. Indeed, the map-labellings of Figure 6(c) and Figure 6(b) differ by very few label positions.

At first thought, the map-labelling of Figure 6(c) may appear of little benefit. However, map-labellings with conflicts can be beneficial in a number of situations. First, for sight-impaired readers, compromising on font-size may not be possible and hence conflicts may need to be tolerated. Alternatively, map-labellings with conflicts may also be useful when we consider partially-labelled maps. Consider the



(a) Conflicts: 0 Font-size: 12pt<sup>†</sup> ROD: 5.92



(b) Conflicts: 0 Font-size: 17pt<sup>†</sup> ROD: 12.01



(c) Conflicts: 4 Font-size: 17pt<sup>†</sup> ROD: 8.95

**Figure 6: Different map-labellings for Run 1 from Figure 3 of our MOEA map-labelling system for the map of Guam. (b) represents the final non-dominated map-labelling with the largest font-size that contains no conflicts. (a) represents a map-labelling with better clarity obtained by compromising font-size. (c) represents a map-labelling with better clarity obtained by compromising the number of conflicts.**

<sup>†</sup>This map has been scaled to fit page requirements; text may not appear at quoted font-size.

map-labelling shown in Figure 6(c) with the Joatan label removed. Using this partially-labelled map, the location of Lingae is immediately apparent as the Lingae label is located very near to the correct site. This is not the case in Figure 6(b) — as described earlier, some deductive work is required to correctly determine the true site for the Lingae label in this map-labelling. The use of partially labelled maps may be particularly useful in online map-generation, where a user may only be interested in a few sites — the algorithm may be able to present a better-clarity partially-labelled map than possible when considering a zero-conflict fully labelled map.

## 4.2 Experiment 2

In this experiment, we report the results of the MOEA map-labelling system for the map of Newcastle County in Delaware, run for 2500 generations. The MOEA attempts to optimise for three font-sizes: 8pt to 10pt inclusive.

Figure 7 plots two non-dominated map-labellings for one run of our MOEA map-labelling system for the map of Newcastle County in Delaware. Figure 7(a) plots the final non-dominated map-labelling with the largest font-size that contains no conflicts. Figure 7(b) plots the final non-dominated map-labelling with the best clarity that contains no conflicts.

Since both map-labellings are non-dominated and contain no conflicts, Figure 7(a) (which has a larger font-size than Figure 7(b)) must have worse clarity than Figure 7(b). Comparing Figure 7(a) to Figure 7(b), we see a number of differences between the two map-labellings than can be attributed to the difference in clarity between the two map-labellings. These regions are highlighted via bolded ellipses in Figure 7(a). Examining these regions in Figure 7(a), we observe local ambiguity about the attachment of labels to sites that requires some deductive work to determine the correct map-labelling. The corresponding regions in Figure 7(b) are much less ambiguous — labels are more clearly attached to their correct site and hence easier to determine from casual inspection. Overall, the map of Figure 7(b) has better clarity than Figure 7(a).

## 5. CONCLUSIONS AND FUTURE WORK

We have presented a multi-objective evolutionary algorithm approach to the map-labelling problem. Being multi-objective, the approach returns a range of solutions that present different compromises between the three criteria of maximising font-size, maximising the clarity of bindings, and minimising the number of conflicts on the map. The approach finds automatically the largest font-size that can give zero conflicts for a map, and it maximises clarity at that font-size. The set of solutions returned also allows users to select between other labellings that sacrifice one objective in favour of the others: for example, accepting a smaller font-size to attain greater clarity, or accepting a map where some labels overlap to attain greater clarity at large font-sizes. This approach is a significant step towards improving the usefulness of fully-automatic map-labelling systems.

Future work in this area will concentrate on three areas:

1. Currently, our map-labelling system proceeds by constraining all labels in a map-labelling to use the same font-size. As seen in the experiments above, this may induce local ambiguities in regions of the map, in particular at larger font-sizes. Alternatively, a representa-

tion that uses a region specific (or label specific) font-size can be used, thus allowing each region (or label) to optimise the font-size for its surrounding neighbourhood, potentially reducing ambiguity in the map.

2. The current clarity objective is defined entirely geometrically, in terms of the relative distances between sites and labels. An alternative might be to define an objective that reflects the logical process that users employ to disambiguate closely-spaced labels.
3. We used a modified definition of dominance to reflect the perceived importance of avoiding conflicts relative to the other two objectives, as defined in Section 3.3. This is probably the most innovative feature of the algorithm, and we intend to study its effects and determine how it can be generalised to other applications.

## 6. ACKNOWLEDGMENTS

We thank Shih Ching Fu from The University of Western Australia for his contributions to this work.

## 7. REFERENCES

- [1] L. Barone, L. While, and P. Hingston. Designing crushers with a multi-objective evolutionary algorithm. In *GECCO 2002*, pages 995–1002. Morgan Kaufmann Publishers, 2002.
- [2] L. Bradstreet. Labelling maps with multi-objective evolutionary algorithms. Honours thesis, The University of Western Australia, 2004.
- [3] D. Büche, P. Stoll, R. Dornberger, and P. Koumoutsakos. Multiobjective evolutionary algorithm for the optimization of noisy combustion processes. *IEEE Transactions on Systems, Man, and Cybernetics — Part C: Applications and Reviews*, 32(4):460–473, 2002.
- [4] S. Cavalieri and P. Gaiardelli. Hybrid genetic algorithms for a multiple-objective scheduling problem. *Journal of Intelligent Manufacturing*, 9(4):361–367, 1998.
- [5] J. Christensen, J. Marks, and S. Shieber. Placing text labels on maps and diagrams. In P. Heckbert, editor, *Graphics Gems IV*, pages 497–504. Academic Press, Boston, MA, 1994.
- [6] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In M. Schoenauer et al., editors, *PPSN VI*, pages 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.
- [7] D. E. Goldberg. *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison-Wesley, 1989.
- [8] S. A. Hirsch. An algorithm for automatic name placement around point data. *The American Cartographer*, 9(1):5–17, 1982.
- [9] E. Imhof. Positioning names on maps. *The American Cartographer*, 2(2):128–144, 1975.
- [10] P. Moscato. Memetic algorithms. In P. Pardalos and M. Resende, editors, *Handbook of Applied Optimization*. Oxford University Press, 2002.
- [11] M. Preuß. Solving map labeling problems by means of evolution strategies. Masters thesis, Fachbereich Informatik, Universität Dortmund, Feb. 1998.
- [12] G. Raidl. A genetic algorithm for labeling point features. In *Proc. International Conference on Imaging Science, Systems, and Technology*, 1998.
- [13] V. S. Summanwar, V. K. Jayaraman, B. D. Kulkarni, H. S. Kusumakar, K. Gupta, and J. Rajesh. Solution of constrained optimization problems by multi-objective genetic algorithm. *Computers and Chemical Engineering*, 26(10):1481–1492, 2002.
- [14] U.S. Department of the Interior. State and topical gazetteer download page. [Online] <http://geonames.usgs.gov/stategaz/index.html>, [16 September 2004].
- [15] S. van Dijk, D. Thierens, and M. de Berg. On the design of genetic algorithms for geographical applications. In W. Banzhaf et al., editors, *GECCO'99*, pages 188–195. Morgan Kaufmann, July 1999.



(a) Conflicts: 0 Font-size: 10pt<sup>‡</sup> ROD: 64.26



(b) Conflicts: 0 Font-size: 8pt<sup>‡</sup> ROD: 44.42

Figure 7: Different map-labellings for the map of Newcastle County in Delaware for one run of our MOEA map-labelling system. (a) represents the final non-dominated map-labelling with the largest font-size that contains no conflicts. (b) represents the final non-dominated map-labelling with the best clarity that contains no conflicts.

<sup>‡</sup>This map has been scaled to fit page requirements; text may not appear at quoted font-size.