An Evolutionary/Meta-Heuristic Approach To Emergency Resource Redistribution In The Developing World

Andrew Tuson, Richard Wheeler and Peter Ross
OUTLOOK Group, Department of Artificial Intelligence,
University of Edinburgh, 80 South Bridge, Edinburgh EH1 1HN
{andrewt,richardw,peter}@dai.ed.ac.uk

Abstract

The problem of logistics and resource management in disease control projects in the developing world can hardly be understated. One problem that arises is the occurance of regional imbalances in supply. Such imbalances can seriously compromise the effectiveness of such programs. A prototype system, based upon evolutionary and 'meta-heuristic' optimisation techniques is described. This system recommends a plan for the redistribution of available resources to minimise shortages for real situations in the developing world. Evaluation of the system indicated that the generation of good, feasible redistribution plans is possible. A comparison of the optimisers showed that evolutionary techniques perform poorly on this problem when compared to stochastic hillclimbing.

1 Introduction

Dealing with aid projects in the developing world can be made difficult by the fact that though an effective policy exists, the actual implementation is often more difficult. For instance, situations may arise where regional imbalances arise. Such imbalances can seriously compromise the effectiveness of a relief program; an example would be a treatment site with plenty of diagnostic kits for a particular disease, but with no drugs to treat the diagnosed cases.

This problem can be described as follows: we have N sites, each of which need to have minimum amounts of M resources in order to operate effectively. Furthermore, each site has different requirements for each resource, and shortages and surpluses of each resource occur at different sites. A resource management system has to give a list of recommendations of the form 'move X amount of resource Z from site A (which has a surplus) to site B (which has a shortage)'. The aim is then to maximise the number of resource targets met (referred to as resource-sites), whilst minimising the number of shipments. This is subject to various constraints on feasibility, for example, a site cannot supply more of a resource than it has.

The hardware limitations are also severe — the computer technology available is hardly state-of-the-art. Therefore the final system aims to run on a 386-class PC compatible, with a 1 MB memory, and a 20 MB hard drive. The goal is to generate plans for regions with at least 250 relief centres within several hours.

This paper outlines a prototype resource management system to evaluate the feasibility of a system that recommends an *implementable* shipment plan to minimise such shortages in real-world situations.

2 Implementation

The above problem was viewed as a constrained optimisation problem: find a redistribution plan that minimises shortages and the number (and cost) of shipments required, whilst satisfying the constraints outlined earlier. The solution adopted consisted of two parts: the representation, which describes the redistribution plan and contains the domain knowledge; and a search algorithm to optimise the plan.

2.1 Representation

The representation of the redistribution plan adopted was indirect [1] — the candidate solution encodes not the redistribution plan; but rather instructions to a plan builder. This approach is compact in its memory usage and always generates feasible plans. This 'anytime' characteristic is of particular use as an answer may be required quickly, and a sub-optimal but feasible answer is preferable to no answer at all. The encoding is the order in which sites are considered for supply (i.e. a permutation) by the plan builder. Details of the plan-builder used are given in [2].

2.2 Optimisation

A variety of hillclimbers and 'meta-heuristic' optimisers were implemented for the optimisation of the initial redistribution plan.

- **Hillclimbing**. Steepest-ascent (SAHC), First-ascent (FAHC), and Stochastic (SHC); with and without iteration.
- **Tabu Search** (TS) [3].
- **Simulated Annealing** (SA) [4].
- **Threshold Accepting** (TA) [5].
- **Evolutionary Algorithm** (EA) [6].

The neighbourhood operators used by all of the optimisation methods for the encoding described were: permutation-shift and permutation-swap [7]. For the Evolutionary Algorithm (see below) the crossover operator used was the 'Modified PMX' operator [8], which has been found to give good results on some permutation-encoded scheduling problems.

The evaluation function used in this system was a simple one — a linear combination of the number of targets met (*targets_met*) and the number of shipments made (*shipments*); as is shown in the following equation.

$$fitness = 10 \times targets_met - shipments \qquad (1)$$

The number of targets met is weighted, so that the system will optimise *targets_met* , before trying to optimise *shipments*. This was deemed a desirable behaviour by the potential end-users.

3 An Initial Evaluation

The data for the test problems was provided by an expert with experience of the type of situations the system is intended to play a part in, and was based upon actual data. A small domain consisting of 21 sites and 12 resources was examined first. This test dataset had the feature that there were roughly sufficient resources available to supply all of the resource targets.

Stochastic hillclimbing was used to give a rough idea of the expected performance of the system. The initial solution met about 85-92% of resource targets when processed by the plan builder, compared to 32% before the system is run. Further optimisation increased this figure to 90-92%, and managed to reduce the number of shipments required by 20% — from about 125 to about 105 shipments — in less than 1000 evaluations. This took 7 seconds on a SPARC 5, and 5 minutes on a 386-class PC.

4 Which Optimiser?

The performance of each of the optimisers, for a range of technique-specific parameter values was investigated, including the choice of a swap or shift neighbourhood. Fifty runs, each lasting 2000 evaluations were taken in each case (where differences in performance are reported to be significant, this has been ascertained using a t-test).

Due to lack of space, not all of the results obtained can be given here (the reader is directed to [2] for full results). Instead, the best results obtained (in terms of solution quality, and then the number of evaluations to solution) for each technique are summarised in Table 1.

Some points can be made about the behaviour of the optimisers in general. The type of hillclimbing that the optimiser was based upon was found to be important. SHC-based optimisers significantly outperformed FAHC and SAHC based optimisers for both types of neighbourhood; with SHC the method of choice for this problem. The choice of neighbourhood made little difference for SHC and FAHC based optimisers. However, for a SAHC-based optimiser, a swap neighbourhood gave better quality solutions. This was due to swap neighbourhoods being roughly half the size of shift neighbourhoods.

4.1 Hillclimbing

Iterating the hillclimber gave a significant improvement for SHC (shift-only), FAHC (both), and SAHC (swap-only).

4.2 Simulated Annealing

Compared against SHC without iteration, solution quality was increased, but not as much as for iterated hillclimbing. Low initial temperatures were found to be most effective.

4.3 Threshold Accepting

The best results were obtained with a low initial threshold of one. Taking the result with SA into account, it appears that there are gains to be made by allowing a little exploration; but this must be used sparingly.

	Shift Neighbourhood		Swap Neighbourhood	
Algorithm	Quality	Evals. Reqd.	Quality	Evals. Reqd.
SHC	2321.96 (1.15)	1091.36 (519.64)	2321.68 (1.19)	1042.56 (474.51)
SA	2321.64 (1.53)	1045.92 (527.88)	2321.26 (1.59)	1074.68 (534.29)
TA	2321.82 (1.19)	981.36 (501.17)	2321.58 (1.30)	1053.26 (467.23)
EA	2320.42 (1.17)	1683.02 (247.80)	2319.98 (1.10)	1526.56 (296.02)
FAHC	2319.86 (1.76)	1083.26 (505.30)	2319.98 (1.41)	953.40 (587.77)
SAHC	2317.46 (1.88)	1921.00 (178.89)	2319.52 (1.99)	1404.02 (461.24)
TS(FA)	2319.16 (2.35)	834.26 (427.57)	2319.24 (1.69)	800.56 (583.02)
TS(SA)	2318.28 (1.83)	$>$ 2000 (-)	2319.34 (2.00)	1346.98 (388.00)

Table 1: A Summary Of Optimiser Performance

4.4 Evolutionary Algorithm

The EA performed less well than SHC. The choice of population model was found to be important. A steady state model gave the best performance, followed by a (N+N) ES model. Generational models performed worst of all (worse than SAHC). Also, crossover seemed to have no positive effect, and in some cases a high crossover probability degraded solution quality slightly. Although it is unclear whether this lies with the type of recombination operator used, or with population-based search in general.

4.5 Tabu Search

The level of performance was not significantly affected by the length of the tabu list, remaining comparable to the basic hillclimber. The exception was SAHC-based tabu search with a shift-neighbourhood; probably due to the tabu list reducing the size of a large neighbourhood.

5 Additional Constraints

So far, the assumption has been made that the shipment can travel over any distance — this is in fact not true. For the real-world problem this system is aimed at, traversing over the catchment area of more than 3 sites is infeasible.

To see if imposing a constraint on shipment distance would affect the coverage attainable, SHC was used with the constraint that the shipments could not be made further than a maximum value of 3. The coverage attained was around 90%, a slight drop compared with 96% without the constraint.

It would also be desirable for the distance of the shipments to be minimised. The evaluation function was modified to include the total *distance* covered by the shipments made.

$$fitness = 10 \times targets_met - distance \qquad (2)$$

The use of the modified evaluation function was found to have no effect upon coverage. A lower distance solution was also attained, though at the expense of using more shipments.

6 Using A Large Dataset

The experiments so far have been performed on a relatively small domain. A practical system will have to deal with more than this. So a larger data-set was used, consisting of 283 sites and 12 resources, based on actual data as before.

Initially, only 19% of the resource targets were met. The initial solution met about 54% of the targets when processed by the plan builder with the constraint that a shipment cannot go further than distance 3. SHC was used with the first of the evaluation functions for 2000 evaluations (which took about 10.5 minutes on a SPARC 5 workstation, or 7-8 hours on a 386-class PC). This increased the number of resource targets met to 57%. The search was then extended for 20,000 evaluations. This gave a best solution of 58%; though the local optimum was not reached at that stage.

The lower coverage obtained, compared with the small dataset, was found not to be due to any failing of the system itself, but that at least 30% of sites in the data set were inaccessible to any of the sites with surpluses. This was due to a combination of the sites with surpluses being heavily concentrated (spatially) at one end of the dataset, and the maximum shipment distance constraint.

7 Extensions

The problem studied still requires more work to fully reflect the real-world situation; addressing this requires discussion with the end-users to ascertain their requirements. Also, the introduction of directed neighbourhood operators and 'delta-evaluation' [9]

could increase the speed of search. Finally, the question of why the EA performed so poorly could be explored. A possible reason may lie in the fact that the crossover operator used manipulates the wrong type of 'building block'; finding out what the appropriate building blocks of this problem are and devising a crossover operator to exploit them may improve EA performance.

8 Conclusion

The system described was able to quickly produce a workable redistribution plan for both small and full-sized versions of the problem which increased dramatically the number of resource targets met. Further optimisation improved upon this further; but this was found to take some time for large problems. Fortunately the 'anytime' characteristic of the system means that this is not a real concern. Experiments with the full-sized dataset indicated that the constraints upon the problem may often prevent all the resource targets being met, even if there are sufficient resources available.

Stochastic hillclimbing was found to be the method of choice, with type of neighbourhood operator used being unimportant. In fact SHC-based optimisers were the best performers; though the performance of the Evolutionary Algorithm was found to be disappointing.

Although more work needs to be performed, these results bode well for attempts to scale up the system to a real-world situation.

9 Acknowledgements

We would like to express our gratitude to the Engineering and Physical Sciences Research Council (EPSRC) for their support of Andrew Tuson via a research studentship (95306458).

References

[1] E. Burke, D. Elliman, and R. Weare. The Automated Timetabling of University Exams using a Hybrid Genetic Algorithm. In *The AISB Workshop on Evolutionary Computing*, 1995.

[2] A. Tuson, R. Wheeler, and P. Ross. A Prototype Resource Management System For The Third World. Working Paper, Department of Artificial Intelligence, Edinburgh University, 1996.

[3] F. Glover. Tabu Search: A Tutorial. *Interfaces*, 4:445–460, 1990.

[4] S. Kirkpatrick, C.D. Gelatt, Jr., and M.P. Vecchi. Optimization by Simulated Annealing. *Science*, 220:671–680, 1983.

[5] G. Dueck and T. Scheuer. Threshold Accepting: A General Purpose Optimisation Algorithm Superior to Simulated Annealing. *Journal of Computation Physics*, 90:161–175, 1990.

[6] J.H. Holland. *Adaptation in Natural and Artificial Systems*. Ann Arbor: The University of Michigan Press, 1975.

[7] C.R. Reeves. A genetic algorithm for flowshop sequencing. *Computers & Ops. Res.*, 22:5–13, 1995.

[8] G.F. Mott. Optimising Flowshop Scheduling Through Adaptive Genetic Algorithms. Chemistry Part II Thesis, Oxford University, 1990.

[9] P. Ross, D. Corne, and H-L. Fang. Improving evolutionary timetabling with delta evaluation and directed mutation. In Y. Davidor, H-P. Schwefel, and R. Manner, editors, *Parallel Problem-solving from Nature - PPSN III*, LNCS, pages 566–565. Springer-Verlag, 1994.