

---

# A Pyramidal Evolutionary Algorithm with Different Inter-Agent Partnering Strategies for Scheduling Problems

---

*in GECCO 2001: Proceedings of the Genetic and Evolutionary Computation Conference, late-breaking papers volume, pp 1-8, San Francisco, USA, 2001.*

Uwe Aickelin  
School of Computer Science  
University of Nottingham  
NG8 1BB UK  
uxa@cs.nott.ac.uk

## Abstract

This paper combines the idea of a hierarchical distributed genetic algorithm with different inter-agent partnering strategies. Cascading clusters of sub-populations are built from bottom up, with higher-level sub-populations optimising larger parts of the problem. Hence higher-level sub-populations search a larger search space with a lower resolution whilst lower-level sub-populations search a smaller search space with a higher resolution. The effects of different partner selection schemes amongst the agents on solution quality are examined for two multiple-choice optimisation problems. It is shown that partnering strategies that exploit problem-specific knowledge are superior and can counter inappropriate (sub-) fitness measurements.

## 1 INTRODUCTION

When hierarchically distributed evolutionary algorithms are combined with multi-agent structures a number of new questions become apparent. One of these questions is addressed in this paper: the issue of intelligently selecting mating partners from another population for an agent. This paper will look at seven different partnering strategies when combined with a genetic algorithm that uses a co-operative sub-population structure. We will evaluate the different strategies according to their optimisation performance of two scheduling problems.

Genetic algorithms are generally attributed to Holland [1976] and his students in the 1970s, although evolutionary computation dates back further (refer to Fogel [1998] for an extensive review of early approaches). Genetic algorithms are stochastic meta-heuristics that mimic some features of natural evolution.

Canonical genetic algorithms were not intended for function optimisation, as discussed by De Jong [1993]. However, slightly modified versions proved very successful. For an introduction to genetic algorithms for function optimisation, see Deb [1996].

The twist when applying our type of distributed genetic algorithm lies in its special hierarchical structure. All sub-populations follow different (sub-) fitness functions, so in effect only searching specific parts of the solution space. Following special crossover-operators these parts are then gradually merged to full solutions. The advantage of such a divide and conquer approach is reduced epistasis within the lower-level sub-populations which makes the optimisation task easier for the genetic algorithm.

The paper is arranged as follows: the following section describes the nurse scheduling and tenant selection problems. Pyramidal genetic algorithms and their application to these two problems are detailed in section 3. Section 4 explains the seven partnering strategies examined in the paper and section 5 describes their use and computational results. The final section discusses all findings and draws conclusions.

## 2 THE NURSE SCHEDULING AND TENANT SELECTION PROBLEMS

Two optimisation problems are considered in this paper, the nurse scheduling problem and the tenant selection problem. Both have a number of characteristics that make them an ideal testbed for the enhanced genetic algorithm using partnering strategies. Firstly, they are both in the class of NP-complete problems [Johnson 1998, Martello & Toth 1990], hence they are challenging problems. Secondly, they have proved resilient to optimisation by a

standard genetic algorithm, with good solutions only found by using a novel strategy of indirectly optimising the problem with a decoder based genetic algorithm [Aickelin & Dowsland 2001]. Finally, both problems are similar multiple-choice allocation problems. For the nurse scheduling, the choice is to allocate a shift-pattern to each nurse, whilst for the tenant selection it is to allocate an area of the mall to a shop. However, as the following more detailed explanation of the two will show, the two problems also have some very distinct characteristics making them different yet similar enough for an interesting comparison of results.

The nurse-scheduling problem is that of creating weekly schedules for wards of up to 30 nurses at a major UK hospital. These schedules have to satisfy working contracts and meet the demand for given numbers of nurses of different grades on each shift, whilst at the same time being seen to be fair by the staff concerned. The latter objective is achieved by meeting as many of the nurses' requests as possible and by considering historical information to ensure that unsatisfied requests and unpopular shifts are evenly distributed. Due to various hospital policies, a nurse can normally only work a subset of the in total 411 theoretically possible shift-patterns. For instance, a nurse should work either days or nights in a given week, but not both. The interested reader is directed to Aickelin & Dowsland [2000] and Dowsland [1998] for further details of this problem.

For our purposes, the problem can be modelled as follows. Nurses are scheduled weekly on a ward basis such that they work a feasible pattern with regards to their contract and that the demand for all days and nights and for all qualification levels is covered. In total three qualification levels with corresponding demand exists. It is hospital policy that more qualified nurses are allowed to cover for less qualified one. Infeasible solutions with respect to cover are not acceptable. A solution to the problem would be a string, with the number of elements equal to the number of nurses. Each element would then indicate the shift-pattern worked by a particular nurse. Depending on the nurses' preferences, the recent history of patterns worked, and the overall attractiveness of the pattern, a penalty cost is then allocated to each nurse-shift-pattern pair. These values were set in close consultation with the hospital and range from 0 (perfect) to 100 (unacceptable), with a bias to lower values. The sum of these values gives the quality of the schedule. 52 data sets are available, with an average problem size of 30 nurses per ward and up to 411 possible shift-patterns per nurse.

For comparison, all data sets were solved using a standard IP package [Fuller 1998]. However, some remained unsolved after each being allowed 15 hours run-time on a Pentium II 200. Experiments with a number of descent

methods using different neighbourhoods, and a standard simulated annealing implementation, were even less successful and frequently failed to find feasible solutions. The most successful approach to date is based on Tabu Search [Dowsland 1998]. However, the quality of solutions relies heavily on data-specific chains of moves that work well because of the way in which the different factors affecting the quality of a schedule are combined. A straightforward genetic algorithm approach failed to solve the problem [Aickelin & Dowsland 2000]. The best evolutionary results to date have been achieved with an indirect genetic approach employing a decoder function [Aickelin & Dowsland 2001]. However, we believe that there is further leverage in direct evolutionary approaches to this problem. Hence we propose to use an enhanced pyramidal genetic algorithm in this paper.

The second problem is a mall layout and tenant selection problem; in future mall problem for short. The mall problem arises both in the planning phase of a new shopping centre and on completion when the type and number of shops occupying the mall has to be decided. To maximise revenue a good mixture of shops that is both heterogeneous and homogeneous has to be achieved. Due to the difficulty of obtaining real-life data because of confidentiality, the problem and data used in this research are constructed artificially, but closely modelled after the actual real-life problem as described for instance in Bean et al. [1988]. In the following, we will briefly outline our model.

The objective of the mall problem is to maximise the rent revenue of the mall. Although there is a small fixed rent per shop, a large part of a shop's rent depends on the sales revenue generated by it. Therefore, it is important to select the right number, size and type of tenants and to place them into the right locations to maximise revenue. As outlined in Bean et al. [1988], the rent of a shop depends on the following factors:

- The attractiveness of the area in which the shop is located.
- The total number of shops of the same type in the mall.
- The size of the shop.
- Possible synergy effects with neighbouring similar shops, i.e. shops in the same group (not used by Bean et al.).
- A fixed amount of rent based on the type of the shop and the area in which it is located.

This problem can be modelled as follows: Before placing shops, the mall is divided into a discrete number of locations, each big enough to hold the smallest shop size. Larger sizes can be created by placing a shop of the same

type in adjacent locations. Hence, the problem is that of placing  $i$  shop-types (e.g. menswear) into  $j$  locations, where each shop-type can belong to one or more of  $l$  groups (e.g. clothes shops) and each location is situated in one of  $k$  areas. For each type of shop there will be a minimum, ideal and maximum number allowed in the mall, as consumers are drawn to a mall by a balance of variety and homogeneity of shops.

The size of shops is determined by how many locations they occupy within the same area. For the purpose of this study, shops are grouped into three size classes, namely small, medium, and large, occupying one, two, and three locations in one area of the mall respectively. For instance, if there are two locations to be filled with the same shop-type within one area, then this will be a shop of medium size. If there are five locations with the same shop-type assigned in the same area, then they will form one large and one medium shop etc. Usually, there will also be a maximum total number of small, medium and large shops allowed in the mall.

To test the robustness and performance of our algorithms thoroughly on this problem, 50 problem instances were created. All problem instances have 100 locations grouped into five areas. However, the sets differ in the number of shop-types available (between 50 and 20) and in the tightness of the constraints regarding the minimum and maximum number of shops of a certain type or size. Full details on how the data was created, its dimensions, the differences between the sets can be found in [Aickelin 1999].

### 3 PYRAMIDAL GENETIC ALGORITHMS

Both problems failed to be optimised with a standard genetic algorithm [Aickelin & Dowsland 2000 and 2001]. Our previous research showed that the difficulties were attributable to epistasis created by the constrained nature of the optimisation. Briefly, epistasis refers to the ‘non-linearity’ of the solution string [Davidor 1991], i.e. individual variable values which were good in their own right, e.g. a particular shift / location for a particular nurse / shop formed low quality solutions once combined. This effect was created by those constraints that could only be incorporated into the genetic algorithm via a penalty function approach. For instance, most nurses preferred working days; thus partial solutions with many ‘day’ shift-patterns have a higher fitness. However, combining these shift-patterns leads to shortages at night and therefore infeasible solutions. The situation for the mall problem is similar yet more complex, as two types of

constraints have to be dealt with: size constraints and number constraints.

In Aickelin & Dowsland [2000] we presented a simple and on its own unsuccessful pyramidal genetic algorithm for the nurse-scheduling problem. A pyramidal approach can best be described as a hierarchical distributed genetic algorithm where cascading clusters of sub-populations are built from bottom up, with higher-level sub-populations optimising larger parts of the problem. Thus, the hierarchy is not within one string but rather between sub-populations which optimise different string-portions. Hence, higher-level sub-populations search a larger search space with a lower resolution whilst lower-level sub-populations search a smaller search space with a higher resolution. This can be applied to the nurse-scheduling problem in the following way:

- Agents in sub-populations 1, 2 and 3 have their fitness based on cover and requests only for grade 1, 2 and 3 respectively.
- Agents in sub-populations 4, 5 and 6 have their fitness based on cover and requests for grades (1+2), (2+3), (3+1).
- Agents in sub-population 7 optimise cover and requests for (1+2+3).
- Agents in sub-population 8 solve the original (all) problem.

The full structure is illustrated in figure 1. Sub-solution strings from lower populations are cascaded upwards using suitable crossover and selection mechanisms. For instance, fixed crossover points are used such that an agent from sub-population (1) combined with one from (2+3) forms a full solution. Although the full problem is as epistatic as before, the sub-problems are less so as the interaction between nurse grades is (partially) ignored. Compatibility problems of combining the parts are reduced by the pyramidal structure with its hierarchical and gradual combining.

Using this approach improved solution quality in comparison to a standard genetic algorithm was recorded. However, the quality of solutions was still short of those produced by Tabu Search. So far roulette wheel selection based on fitness rank had been used to choose parents. The fitness of each sub-string is calculated using a substitute fitness measure based on the requests and cover as detailed above, i.e. the possibility of more qualified nurses covering for less-qualified ones is partially ignored. Unsatisfied constraints are included via a penalty function. This paper will investigate various partnering strategies between the agents of the sub-populations to improve upon these results.

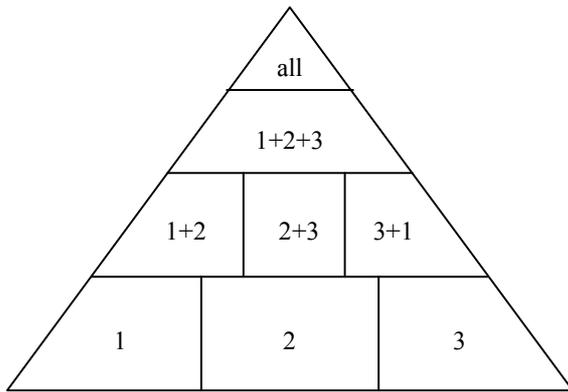


Figure 1: Nurse Problem Pyramidal Structure.

Similar to the nurse problem, a solution to the mall problem can be represented by a string with as many elements as locations in the mall. Each element then indicates what shop-type is to be located there. The mall is geographically split into different regions, for instance north, east, south, west and central. Some of the objectives are regional; e.g. the size of a shop, the synergy effects, the attractiveness of an area to a shop-type, whereas others are global, e.g. the total number of shops of a certain type or size.

The application of the pyramidal structure to the mall problem follows along similar lines to that of the nurse problem. In line with splitting the string into partitions with nurses of the same grade, the string is now split into the areas of the mall. Thus, we will have sub-strings with all the shops in one area in them. These can then be combined to create larger ‘parts’ of the mall and finally full solutions.

However, the question arises how to calculate the substitute fitness measure of the partial strings. The solution chosen here will be a pseudo measure based on area dependant components only, i.e. global aspects are not taken into account when a substitute fitness for a partial string is calculated. Thus, sub-fitness will be a measure of the rent revenue created by parts of the mall, taking into account those constraints that are area based. All other constraints are ignored. A penalty function is used to account for unsatisfied constraints.

Due to the complexity of the fitness calculations and the limited overall population size, we refrained from using several levels in the hierarchical design as we did with the nurse scheduling. Instead a simpler two level hierarchy is used as shown in figure 2: Five sub-populations optimising the five areas separately (1,2,3,4, 5) and one main population optimising the original problem (all). A special crossover then selects one solution from each sub-

population and pastes them together to form a full solution.

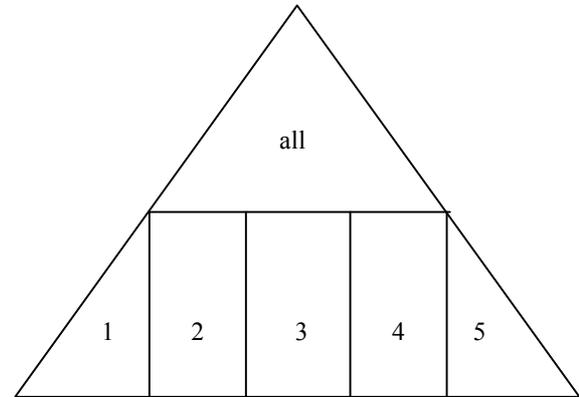


Figure 2: Mall Problem Pyramidal Structure.

The remainder of this paper will investigate ways to try to improve on previously found poor results by suggesting ways of combining partial strings more intelligently. An alternative, particularly for the mall problem, would be a more gradual build-up of sub-populations. Without increasing the overall population size, this would lead to more and hence smaller sub-populations. However, this more gradual approach might have enabled the algorithm to find good feasible solutions by more slowly joining together promising building blocks. This is in contrast to the relatively harsh two-level and three-level design where building blocks had to ‘succeed’ immediately. Exploring the exact benefits of a gradual build-up of sub-solutions would make for another challenging area of possible future research.

## 4 PARTNERING STRATEGIES

The problem of how to pick crossover partners has been noted in both competitive and co-operative co-evolutionary algorithms. Many strategies have been presented in the literature as summarised for instance by Bull [1997]. In this paper seven such strategies are compared for their effectiveness in fighting epistasis in the pyramidal genetic algorithm optimising the nurse scheduling and the mall problems.

Rank-Selection (S): This is the method used so far in our algorithms. Agents are assigned a sub-fitness score based as closely as possible on the contribution of their partial string to full solutions. All agents are then ranked within each sub-population and selection follows a roulette

wheel scheme based on the ranks [e.g. Aickelin & Dowsland 2000].

Random (R): Agents choose their mating partners randomly from amongst all those in the sub-population their sub-population is paired with [e.g. Bull & Fogarty 1993].

Best (B): In this strategy each agent is paired with the currently best agent of the other sub-population(s). In case of a tie, the agent with the lower population index is chosen [e.g. Potter & De Jong 1994].

Distributed (D): The idea behind this approach is to match agents with similar ones to those paired with previously [e.g. Ackely & Littman 1994]. To achieve this each sub-population is spaced out evenly across a single toroidal grid. Subsequently, agents are paired with others on the same grid location in the appropriate other sub-populations. Children created by this are inserted in an adjacent grid location. This is said to be beneficial to the search process because a consistent co-evolutionary pressure emerges since all offspring appear in their parents' neighbourhoods [Husbands 1994]. In our algorithms we use local mating with the neighbourhood set to the eight agents surrounding the chosen location.

Joined (J): In nature, some species carry others internally with the relationship propagated from generation to generation [e.g. Iba 1996]. Thus, each agent represents a complete solution; i.e. all the parts have been joined together. In our case, this results in all sub-populations solving the original problem, i.e. we have a traditional parallel genetic algorithm. This means that all sub-populations use the full fitness function for evaluation and rank-proportional selection.

Attractiveness (A): The five strategies described so far are general and do not make use of problem specific knowledge. However, there is a growing body of research [e.g. Stanley et al. 1994, Wolpert & Macready 1995], as well as our own previous work, which suggests that approaches that exploit problem specific knowledge achieve better results. Here pairing is done as for the rank-selection strategy (S). However, the pair is only accepted with a probability proportional to their fitness or substitute fitness once combined. The probabilities are scaled such that if the (substitute) fitness  $f_{comb}$  is equal or greater to the best-known fitness  $f_{best}$  the pairing is automatically accepted. Otherwise the probability is  $f_{comb} / f_{best}$  for the mall problem and the inverse for the nurse scheduling.

Partner Choice (C): This approach again exploits problem specific knowledge and was inspired by an idea presented

by Ronald [1995]. He solves Royal Roads and multi-objective optimisation problems using a genetic algorithm where the first parent is chosen following standard rules, i.e. proportional to its fitness. However, the second parent is not chosen according to its fitness, but depending on its 'attractiveness' to the first parent, which is measured on a different scale. Our approach will be slightly different. The first parent is still chosen according to its rank. But rather than picking one agent from the appropriate sub-population as the second parent, ten candidates are chosen at random. The second parent will then be chosen as the one that creates the fittest children with the first parent.

## 5 EXPERIMENTAL RESULTS

### 5.1 THE MODEL

To allow for fair comparison, the parameters and strategies used for both problems are kept as similar as possible. Both have a total population of 1000 agents. These are split into sub-populations of size 100 for the lower-levels and a main population of size 300 for the nurse scheduling and respectively of size 500 for the mall problem. In principle, two types of crossover take place: within sub-populations a two-parent-two-children parameterised uniform crossover with  $p=0.66$  for genes coming from one parent takes place.

Between sub-populations a fixed-point crossover is used such that appropriate parts are assembled. For instance, in the nurse problem agents of sub-population 1 and 2 would parent a new child for sub-population 1+2. All in all 50% of children are created via the uniform and the remainder with the fixed-point crossover. For some of the fixed point cases a choice exists, e.g. in the nurse problem new agents of sub-population 1+2+3 can be created in four different ways, either (1+2) + (3), (2+3) + (1), (3+1)+(2) or (1)+(2)+(3). In these situations there is an equal probability for each child to be created in either way. Parent selection followed the seven strategies outlined above.

Each new solution created undergoes mutation with a 1% bit mutation probability, where a mutation would re-initialise the bit in the feasible range. The algorithm is run in generational mode to accommodate the sub-population structure better. In every generation the worst 90% of parents of all sub-populations are replaced. For all fitness and sub-fitness function calculations a fitness score as described before is used. Constraint violations are penalised with a dynamic penalty parameter, which adjusts itself depending on the (sub)-fitness difference between the best and the best feasible agent in each (sub-) population. Full details on this type of weight and how it

was calculated can be found in Smith & Tate [1993] and Aickelin & Dowsland [2000]. The stopping criterion is the top sub-population showing no improvement for 50 generations.

To obtain statistically sound results all experiments were conducted as 20 runs over all problem instances. All experiments were started with the same set of random seeds, i.e. with the same initial populations. The results are presented in feasibility and cost respectively rent format. Feasibility denotes the probability of finding a feasible solution averaged over all problem instances. Cost / Rent refer to the objective function value of the best feasible solution for each problem instance averaged over the number of instances for which at least one feasible solution was found.

Should the algorithm fail to find a single feasible solution for all 20 runs on one problem instance, a censored observation of one hundred in the nurse case and zero for the mall problem is made instead. As we are minimising the cost for the nurses and maximising the rent of the mall, this is equivalent to a very poor solution. For the nurse-scheduling problem, the cost represents the sum of unfulfilled nurses' requests and unfavourable shift-patterns worked. For the mall, the values for the rent are in thousands of pounds per year.

## 5.2 RESULTS

Table 1 shows the results found by our algorithms for the two problems (N = Nurse problem, M = Mall problem) using the seven different partnering strategies in combination with the pyramidal structure. The results are compared to those found by the standard genetic algorithm (SGA) [Aickelin & Dowsland 2000 and 2001] and the Tabu Search results [Dowsland 1998] for the nurse problem and theoretical bounds for the mall problem (both referred to as 'bound'). A number of interesting observations can be made.

In the nurse scheduling case, the SGA approach failed to find good or even feasible solutions for many data sets. This can be explained by the high degree of epistasis present and the inability of the unmodified genetic algorithm to deal with it. Once the pyramidal structure with rank-based selection (S) is introduced, results improve significantly, however they are still below those found by Tabu Search. For the mall problem, the situation is different. Results found by the SGA are fairly good with high feasibility. This indicates the higher number of feasible solutions for this problem. Solution quality seems reasonably good, too. However, the addition of the pyramidal structure (S) results in a marked deterioration of results.

How can these different results be explained? With the nurse scheduling, the objective function value of a partial solution was obtained by summing the cost values of the nurses and shift-patterns involved. Furthermore, we were able to define relatively meaningful sub-fitness scores by exploiting the 'cumulative' nature of the covering constraints due to the grade structure. Hence the substitute fitness scores calculated allowed for an effective recombination of partial solutions for the nurse-scheduling problem. Thus, there is a good correlation between the sub-fitness of an agent (and hence its rank and its chance of being selected) and the likelihood that it will form part of a good solution. This also explains why the random (R) scheme produces worse results. The best (B) strategy although giving better results than the random selection fails to solve many problems. However, closer observation of experiments showed that it solved some single data sets well. This indicates that genetic variety is as important as fitness in the evolution of good solutions.

Both the distributed (D) and joint (J) strategies again fail to provide better solutions than the rank-based selection. The distributed strategy is similar to the random strategy as it too ignores fitness scores for selection. Choosing from a fixed pool does have some benefits as the results are better than for complete random choice. The joint strategy works almost as well as the rank-selection. This shows that the principle of the 'dividing and conquering' works well with the nurse problem split along the grade boundaries. The slightly poorer results can be explained by the 'full' evaluation of all sub-strings although only 'parts' are passed on. Thus, some of the correlation described above is lost.

The two best strategies both outperforming (S) are partner selection based on attractiveness (A) and choice (C). Again this further confirms that the partial sub-fitness scores are a good criterion of selection for the pyramidal algorithm. Overall, (C) is better than (A) which corresponds to (C) having a higher selection pressure than (A), which in turn has a higher selection pressure than (S). To conclude, it seems that for this problem a good correlation between agents' sub-fitness, the pyramidal structure and good full solutions exist. Hence, the scheme with the highest selection pressure using most problem specific information scores best. However, the results also show that even this scheme cannot compete with the Tabu Search results, something we will discuss in the concluding section.

With the Mall Problem, the situation is more complicated since unlike for the nurse problem a large part of the objective function is a source of epistasis, which the proposed partitioning of the string will not eliminate fully. The constraints are a second source for epistasis. In

contrast to the objective function, these depend largely on the whole string, as for instance the total number of shops of a particular size allowed. Only after adding up the shops and sizes for all areas is it known if a solution is feasible or not. So unsurprisingly, a combination of these partial solutions is often unsuccessful because it usually violates the overall constraints.

On their own, solutions of the sub-populations are extremely unlikely to be feasible for the overall problem, as they covered only one fifth of the string. It is equally unlikely for those solutions in the main population, which are formed from the five sub-populations, to be feasible. Although these solutions are of high rent, because the sub-populations ignore the main constraints, their combination is unlikely to produce an overall feasible solution.

The situation is only slightly better with those solutions formed by an agent of the sub-populations and an agent of the main population. Usually, even if the agent of the main population is feasible, the children were not. Again, even though the partial string from the sub-population agent was of high rent, it was usually incompatible with the rest of the string, resulting in too many or too few shops of some types. Thus, in contrast to the nurse-scheduling problem, their sub-fitness scores are a far poorer predictor for the compatibility of the parts to form complete solutions.

This is confirmed by the above average performance of the random strategy (R) and the extremely poor results found by the best strategy (B). Similarly to before, the distributed strategy (D) performs well again giving credit to the idea of even selection pressure without relying on fitness scores, whereas the joint strategy (J) performs poorly suffering both from the unsuitable sub-fitness scores and the now hindering pyramidal structure.

Overall, the real winners are again the more complex strategies of choice (C) and attraction (A). At first this seems contradictory as these rely heavily upon the sub-fitness scores. However, apart from the rank-based initial selection of the first parent, subsequent fitness calculations are made after combining the agents. Since the mall pyramid only has two layers, these combinations are always full solution and hence the full fitness score is used. Thus, the direct link between high fitness and good solutions is re-established. Of the two, (A) performs better than (C). This seems to show that a certain amount of randomness is still important here, which again might be an indication for the lower predictive quality of the sub-fitness scores.

Bound	8.8	100%	2640	100%
SGA	54.2	33%	1850	94%
S	17.6	75%	1540	78%
R	37.4	54%	1790	86%
B	27.1	57%	1490	70%
D	26.5	61%	1770	84%
J	19.9	71%	1590	78%
A	12.2	83%	1950	98%
C	11.1	87%	1910	94%

Table 1: Experimental Results (N = Nurse, M = Mall).

## 6 CONCLUSIONS

This paper has shown the effect different partner strategies have on a pyramidal genetic algorithm solving two different optimisation problems from the area of multiple-choice scheduling. The result for the five simple strategies (S, R, B, D and J) differ for both problems. This is a reflection of the accurateness of the sub-fitness measure in the sense of its predictive power for sub-solutions to form full solutions following the pyramidal recombination strategies. Therefore, in the case of the nurse problem with a good match between sub-fitness and usefulness for recombination the simple strategies worked well, whereas for the mall problem with its poorer correlation between the two it did not.

For both problems the distributed partnering strategy (D) gives consistent results. This leads to the question whether some form of fitness sharing might have been beneficial to some or all of the partnering strategies. We are taking up this idea in our ongoing research into pyramidal genetic algorithms.

The two more advanced strategies (A) and (C) use most problem specific knowledge and work well for both problems. They worked well for the nurse problem because the sub-fitness scores are meaningful. They also worked well for the mall problem because the partners are chosen based on a fitness score after recombination, which in this case equals the full original fitness score. Thus, choosing parents ‘post-birth’ after evaluating possible children can overcome possible shortcomings in the sub-fitness measure.

	N Cost	N Feas	M Rent	M Feas
--	--------	--------	--------	--------

It has to be noted that overall none of the partnering schemes managed to outperform the Tabu Search algorithm or come as close to the bounds as we had hoped. However, one has to remember that the Tabu Search uses highly problem-specific hill-climbing routines, some of which also relied heavily on certain criteria present in the actual data used. For different or more random data this would probably no longer hold.

Therefore, without adding a more specific hill-climbing component it would not be possible to reach this level of solution quality. It is our conclusion that pyramidal genetic algorithms (with or without hill-climbing) benefit greatly from the right choice of partnering strategy as this improves solution quality and negates possible shortcoming of the chosen sub-fitness scores.

## References

- Ackley D H & Littman M L (1994), "Altruism in the Evolution of Communication", in R Brooks & P Maes (eds.) *Artificial Life IV*, MIT Press, Mass., pp40-48.
- Aickelin U (1999). "Genetic Algorithms for Multiple-Choice Optimisation Problems." PhD Dissertation, University of Wales, Swansea, United Kingdom.
- Aickelin U and Dowsland K (2000). "Exploiting problem structure in a genetic algorithm approach to a nurse rostering problem." *Journal of Scheduling* 3, 139-153.
- Aickelin U and Dowsland K (2001). "An indirect genetic algorithm approach to a nurse scheduling problem." Under review by the *Journal of Computing and Operational Research*.
- Bean J, Noon C, Ryan S, Salton G (1988), "Selecting Tenants in a Shopping Mall," *Interfaces* 18, 1-9.
- Bull, L. (1997) *Evolutionary Computing in Multi-Agent Environments: Partners*. In T.Baack (ed.) *Proceedings of the Seventh International Conference on Genetic Algorithms*. Morgan Kaufmann, pp370-377.
- Bull L & Fogarty T C (1993), "Coevolving Communicating Classifier Systems for Tracking", in R F Albrecht, C R Reeves & N C Steele (eds.) *Artificial Neural Networks and Genetic Algorithms*, Springer-Verlag, New York, pp522-527.
- Davidor Y. (1991), *Epistasis Variance: A Viewpoint on GA-Hardness*, *Foundations of Genetic Algorithms* 1, 23-35, G Rawlins (Ed), Morgan Kaufmann, 1991.
- De Jong K. Genetic Algorithms are NOT Function Optimisers. In Whitley D. Editor. *Foundations of Genetic Algorithms* 2. San Mateo: Morgan Kaufmann Publishers, 1993:5-17.
- Deb K. *Genetic Algorithms for Function Optimisation*. *Genetic Algorithms and Soft Computing* 1996:4-31.
- Dowsland K.A., Nurse Scheduling with Tabu Search and Strategic Oscillation, *European Journal of Operational Research* 106, 393-407 (1998).
- Fogel D. *Evolutionary Computation: The Fossil Record*. IEEE Press, 1998.
- Fuller E. *Tackling Scheduling Problems Using Integer Programming*. Master Thesis, University of Wales Swansea, United Kingdom, 1998.
- Holland J. *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press, 1976.
- Husbands P (1994), "Distributed Coevolutionary Genetic Algorithms for Multi-Criteria and Multi-Constraint Optimisation", in T C Fogarty (ed.) *Evolutionary Computing*, Springer-Verlag, pp150-165.
- Iba H (1996), "Emergent Co-operation for Multiple Agents Using Genetic Programming", in H-M Voigt, W Ebeling, I Rechenberg & H-P Schwefel (eds.) *Parallel Problem Solving from Nature - PPSN IV*, Springer, Berlin, pp32-41.
- Johnson D.S. private communication 1998.
- Martello S. and Toth P., *Knapsack Problems*, Wiley, Chichester, 1990
- Potter M. & De Jong K. (1994), "A Co-operative Coevolutionary Approach to Function Optimisation", in Y Davidor, H-P Schwefel & R Manner (eds.) *Parallel Problem Solving From Nature - PPSN III*, Springer-Verlag, Berlin, pp249-259.
- Ronald E (1995), *When Selection Meets Seduction*, pp 167-173 in Eshelman L, *Proceedings of the International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, San Francisco, 1995.
- Smith A. and Tate D. (1993), *Genetic Optimisation Using a Penalty Function*, *Proceedings ICGA 5*, 499-505, Forrest S (ed.), Morgan Kaufmann, 1993.
- Stanley A E, Ashlock D, Testatsion L (1994), "Iterated Prisoner's Dilemma with Choice and Refusal of Partners", in C G Langton (ed.) *Artificial Life III*, Addison-Wesley, Redwood City, pp131-146.
- Wolpert D & Macready W (1995), *No Free Lunch Theorem for Search*, SFI-TR-95-02-010, The Santa Fe Institute, Santa Fe, pp 1-38, 1995.