KSATS-HH: A Simulated Annealing Hyper-Heuristic with Reinforcement Learning and Tabu-Search

Kevin Sim

School of Computing
Edinburgh Napier University
Merchiston Campus
Edinburgh EH10 5DT
UK
K.Sim@napier.ac.uk

Abstract. The term hyper-heuristics describes a broad range of techniques that attempt to provide solutions to an array of problems from different domains. This paper describes the implementation of a hyper-heuristic developed for entry into the Cross-domain Heuristic Search Challenge organised by the Automated Scheduling, Optimisation and Planning research group at the University of Nottingham. An algorithm taking inspiration from the Simulated Annealing, Tabu Search and Reinforcement Learning paradigms was implemented and incorporated into the supplied Hyper-heuristics Flexible framework with initial results proving promising, outperforming the benchmark results given.

Keywords: hyper-heuristics, simulated annealing, tabu-search, reinforcement learning.

1 Implementation of a Hyper-Heuristic for the CHeSC Challenge

A number of different approaches ranging from a multi point population based search inspired by an evolutionary algorithm to a simpler greedy iterated local search were implemented in order to get a feel for the HyFlex framework before settling on an approach that uses a Simulated Annealing [4] (SA) strategy whilst incorporating elements inspired by the Tabu-Search [3] (TS) metaheuristic and Reinforcement Learning (RL) methodology.

Figure 1 shows the pseudo code for the implemented algorithm which implements a single point search comprising of two distinct stages, heuristic selection and solution acceptance. Heuristic selection is implemented by means of a tournament between heuristics with each heuristic ranked based on its previous ability to improve the solution. Heuristics that are successful have their rank incremented with a non improving heuristic's rank decremented and the heuristic added to a tabu list for the next 7 iterations. Two heuristics are chosen at

random to be tournament contenders from those heuristics not in the tabu list with the higher ranked of the 2 chosen as the heuristic to be applied. Solution acceptance is taken care of by the SA strategy with solutions accepted as defined by equation 1.

The SA strategy is modified in an attempt to cope with the varied fitness measures used across different problem domains. The change in fitness value, $\Delta\omega$, is normalised by calculating the percentage change in fitness from the current best objective function value allowing different scales of fitness measure to be handled similarly. The cooling schedule is calculated to take account of the average execution time of a heuristic for a given problem instance using equation 3 with $n = 15$ found empirically to be a good all round value for the 4 domains supplied. The temperature at the start of the run, t_0, is set to 1 and reduced at each subsequent iteration using the formula given in equation 2. Equation 4 determines the maximum number of iterations that the algorithm executes without improvement. If this figure is exceeded then the parameters controlling the cooling schedule are reset to their initial values.

$$P = \begin{cases} exp(\frac{-\Delta\omega}{t_k}) & : \Delta\omega > 0 \\ 1 & : \Delta\omega \leq 0 \end{cases} \tag{1}$$

$$t_k = t_{k-1} \times c \quad : 0 < c < 1 \tag{2}$$

$$c = 1 - \frac{1}{time\ taken\ for\ n\ iterations} \tag{3}$$

$$Iter_{max} = \frac{Time\ limit}{time\ taken\ for\ n\ iterations} \tag{4}$$

The approach taken proved competitive, outperforming the 8 sets of supplied results when taken collectively over all 40 problems. Section 2 documents the results obtained with conclusions drawn in section 3.

2 Results

10 runs on each of the 4 problem domains and their associated 10 problem instances were conducted and the results compared to the supplied benchmark results supplied by ASAP. The algorithm developed exhibited varying performance across the 4 domains but when the results were taken collectively and scored using the supplied *formula 1 based scoring system* the results were favorable coming first when contrasted against ASAP's 8 benchmark hyper heuristics using all 40 problems. Figures 2 - 5 show the normalised scores obtained for each domain with the best score for each problem obtained by the organisers superimposed. Table 1 gives the ranked score obtained contrasted against the results supplied by ASAP for 8 benchmark heuristics.

```
– Disregard the crossover heuristics
– Average the time taken to apply heuristics once.
– Calculate the average number of iterations, iterMax,
  to be executed if the algorithm is to be restarted
  numRestarts times.
– Set the Simulated Annealing cooling schedule
  using cooling = 1 − 1/iterMax
– While within the allowed time
–     Choose a heuristic ∉ the tabu heuristics through
      a tournament selection process
–     Determine whether to keep the solution based on
      the Simulated Annealing strategy.
–     If the solution is not better
          Add the heuristic to the tabu heuristics.
          Reset its fitness to min.
       Else
          Increment the heuristics fitness if < max.
–     If There is no improvement for more than iterMax
          Reset the cooling schedule.
      Else
          Reduce the temperature.
–     Release any heuristics that have exceeded the
      iteration limit for the tabu list.
– End While
```

Fig. 1. Pseudo Code

Table 1. Simulated Annealing Tabu Search Hyper-Heuristic Results

KSATS_HH Results					
	SAT	**BPP**	**PS**	**FS**	**Total**
ASAP HH1	53.83	47.00	**66.00**	28.00	194.83
ASAP HH2	71.83	51.00	61.50	19.50	203.83
ASAP HH3	36.50	69.00	22.00	25.00	152.50
ASAP HH4	24.50	59.00	49.50	**81.00**	214.00
ASAP HH5	1.00	10.00	48.00	16.50	75.50
ASAP HH6	46.00	41.00	0.00	74.50	161.50
ASAP HH7	49.83	29.00	50.00	18.50	147.33
ASAP HH8	14.50	1.00	31.00	64.00	110.50
YourHH	**92.00**	**83.00**	62.00	63.00	**300.00**

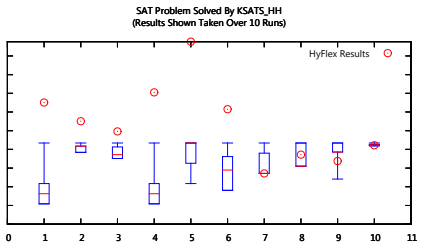

Fig. 2. SAT Results

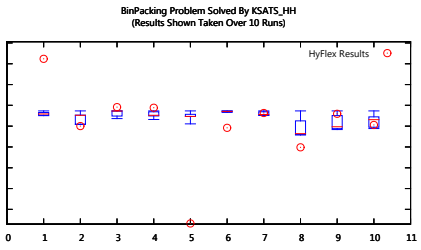

Fig. 3. Bin Packing Results

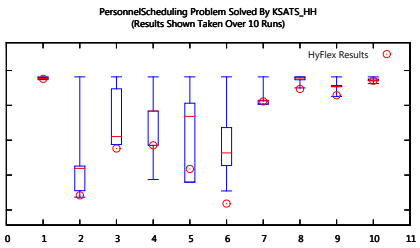

Fig. 4. Personnel Scheduling Results

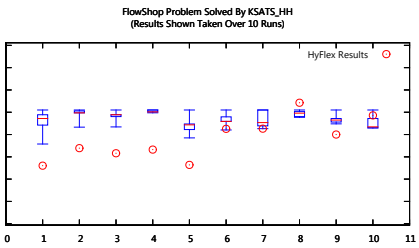

Fig. 5. Flow Shop Results

3 Conclusions

The competition proved a valuable introduction to the emerging field of hyper-heuristics, allowing for investigation of a number of problem domains, domain specific heuristics and metaheuristic techniques. However the system is restricted to using perturbative heuristics with no ability to explore other approaches such as using constructive heuristics or *"heuristics to generate heuristics"*[2].

The approach taken was manually tailored in an attempt to obtain reasonable results across all 4 test domains. Whilst the implementation suffered from the "no free lunch" theorem [5] with modifications which increased performance on one domain having an adverse effect on the algorithm's behavior for other domains it was however able to traverse domains providing "good enough soon enough cheap enough"[1] solutions.

References

1. Burke, E., Kendall, G., Newall, J., Hart, E., Ross, P., Schulenburg, S.: Hyper-Heuristics: An Emerging Direction in Modern Search Technology. In: Handbook of Metaheuristics, chap. 16, pp. 457–474. International Series in Operations Research & Management Science, Kluwer (2003)
2. Burke, E.K., Hyde, M., Kendall, G., Ochoa, G., zcan, E., Woodward, J.R.: A classification of hyper-heuristic approaches. In: Gendreau, M., Potvin, J.Y. (eds.) Handbook of Metaheuristics, International Series in Operations Research &; Management Science, vol. 146, pp. 449–468. Springer US (2010)
3. Glover, F.: Future paths for integer programming and links to artificial intelligence. Comput. Oper. Res. 13, 533–549 (May 1986)
4. Kirkpatrick, S., Gelatt, C.D., Jr, Vecchi, M.P.: Optimization by Simulated Annealing. Science 220, 671–680 (1983)
5. Macready, W.G., Wolpert, D.H.: What makes an optimization problem hard? Complex. 1(5), 40–46 (1996), 228601