International Journal of Soft Computing 8 (2): 108-112, 2013
ISSN: 1816-9503
© Medwell Journals, 2013

Local Search Heuristics for the One Dimensional Bin Packing Problems

Masri Ayob, Mohd Zakree Ahmad Nazri and Yang Xiao Fei
Center for Artificial Intelligence, Faculty of Information Science and Technology,
Universiti Kebangsaan Malaysia, 43600 UKM, Bangi, Selangor, Malaysia

Abstratct: This resaerch implements three basic local search heuristics; hill climbing (i.e., random descent), simulated annealing and multi-start simulated annealing. The aim is to investigate the performance of these heuristics compared to the state of art literatures. To achieve this, researchers use a common software interface (the HyFlex framework) that are designed to enable the development, testing and comparison of iterative general-purpose heuristic search algorithms. To evaluate the performance of these heuristics researchers test on one dimensional bin packing instances using simple move operator. The results demonstrated that hill climbing heuristic outperforms other approaches in all tested instances. This indicates that simple local search is more effective in solving one dimensional bin packing problems when the searcher is allowed to run in a short time.

Key words: Hill climbing, simulated annealing, multi-start simulated annealing, HyFlex, packing

INTRODUCTION

Many real-world problems fall in the class of NP-hard problems (Cook, 1971). That is no algorithm that is able to compute an optimal solution within a timespan that is polynomial in the size of the given problem instance (unless P = NP). In other words, the problem is difficult to solve for optimality in a reasonable time. A good practical example is the problem of one dimensional bin packing problem.

Hence, many researchers have developed variants of efficient heuristic methods to deal with large real-world problem instances. Heuristics (or meta-heuristics) often provide good solutions to NP-hard problems in a reasonable short time but they cannot guarantee to find the optimal solution. Metaheuristics attempt to provide a non problem-specific optimisation algorithm which explores a search space in a guided manner in order to quickly find (near) optimal solutions (Blum *et al*., 2008) for a list of accepted definitions for metaheuristics.

Researchers implemented a wide variety of different metaheuristics approaches such as simulated annealing (Dowsland *et al*., 2007) (modelling a physical cooling process) ant colony optimisation (Dorigo and Gambardella, 1997) (mimicking the way a collection of ants finds a short way to a food source) and Genetic Algorithms (Reeves and Rowe, 2002) (implementing the biological process of gene mutation and recombination). Researchers usually refine their heuristic and often attempt to demonstrate their superiority over the state of art. The hope is to have a universally predominant class of metaheuristics. However, the no free lunch theorems stated that no algorithm can reserach well over divers set of problems (Wolpert and Macready, 1997). Therefore, the meta-heuristic need to be revised to solve a new problem.

The performance of simple iterative improvement local search (such as hill climbing for maximisation problems or descent heuristic in minimisation problems) that iteratively search for better quality solution and only accept improved solutions is in general unsatisfactory (Blum *et al*., 2008). The quality of the solution found in local optimum is heavily depends on the starting point of the local search. A simple strategy to overcome this is by iteratively restart the local search at different starting points which may randomly generated. However, their performances are still far away from being satisfactory (Schreiber and Martin, 1999).

Therefore, this research attempt to further investigate the performance of three basic local search heuristics, simple descent (hill climbing), simulated annealing and multi-start simulated annealing. The hypothesis is if the intelligent mechanism in advanced metaheuristic approaches effect the performance of the heuristic then the advanced metaheuristic approaches will have better performance than basic local searchers.

MATERIALS AND METHODS

The HyFlex framework: HyFlex (Hyper-heuristics Flexible framework) (Ochoa *et al*., 2012) is a Java object oriented framework for the development and comparison of different iterative general-purpose heuristic search

Corresponding Author: Masri Ayob, Center for Artificial Intelligence, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 UKM, Bangi, Selangor, Malaysia

algorithms. HyFlex provides a software interface between the problem specific components and the domain independent algorithm components. The aim is to support researchers in their efforts to develop generally applicable search heuristics. Therefore, future researchers can focus in the design of intelligent and adaptable cross-domain search controllers.

At the highest level, the framework consists of two abstract classes; problem domain and hyper heuristic. HyFlex can be used to implement both population-based and single based metaheuristics/hyper-heuristics . Indeed, it provides six combinatorial optimisation problems with real-world instances data. These are Maximum Satisfiability (MAX-SAT), one-dimensional bin packing, permutation flow shop, personnel scheduling, traveling salesman problems and capacitated vehicle routing problems.

Each domain includes 10 training instances from different sources and a number of problem-specific local search (basic) heuristics of the types. Each HyFlex domain module encapsulates the problem model and the data structure for encoding a candidate solution. It has:

- A routine to initialise randomised solutions
- An objective function for evaluating the quality of solutions
- A set of varied instance data sets from different sources

HyFlex also has various set of low level heuristics which are classified into four groups.

Mutational or pertubation heuristics: These induce a small modification to the current solution.

Ruin-recreate (destruction-construction) heuristics: They operate by randomly destroying part of the solution and then rebuilding it using a greedy or constructive procedure. These heuristics are considered as large neighbourhood structures. They are different from the mutational heuristics in that they can incorporate problem specific construction heuristics to rebuild the solutions.

Hill-climbing (simple descent) heuristics: Operate by iteratively perturbing an incumbent solution accepting improving until a local optimum is found or a stopping condition is met. They differ from mutational heuristics in that they incorporate an iterative improvement process.

Crossover heuristics: Widely used in evolutionary approaches crossover or recombination heuristics that take two solutions and combine them to produce an offspring solution.

Finally, HyFlex provides two control parameters α (intensity of mutation) and β (depth of search) than is used to control the behavior of the heuristics. The precise functioning of the parameters depends on the specific heuristic and problem domain. The value for α and β are between 0 and 1. Details discussion on HyFlex can be referred by Ochoa *et al*. (2012).

HyFlex provides an opportunity to modify some of the heuristics in an informed manner. It is possible to increase or decrease the perturbation effect of a mutational heuristic. In addition to that it is also allowed to change the depth of the search for local search heuristics.

One dimensional bin picking problems: One dimensional packing problem is an NP-hard problem. It consists of a set of pieces that must be packed into the fewest number of bins. This problem has a set of items of a fixed weight and a set of bins of fixed capacity. The packing process must consider these constraints (hard constraint):

- Each item must be assigned to one bin only
- The total weight of items in each bin must be less or equal to the bin capacity

The aim is to minimise to the number of used bins. A mathematical formulation of the classical one dimensional bin packing problem is shown in Eq. 1 (Martello and Toth, 1990):

$$\text{Minimise} \sum_{i=0}^{n} y_i \tag{1}$$

Subject to:

$$\sum_{j=1}^{n} =1 w_j\, x_{ij} \ll c y_i, \quad i \in N = \{1,\ldots,n\}$$

$$\sum_{i=0}^{n} x_{ij} = 1, \quad j \in N$$

$$y_i \in \{0,1\} \quad i \in N, \quad x_{ij} \in \{0,1\} \quad i \in N, j \in N$$

where, $x_{ij} = 1$ indicates that piece j is packed into bin i or otherwise $x_{ij} = 0$; $y_i = 1$ if a bin i contains some pieces, 0 otherwise. N is the number of available bins (and also the number of pieces as researchers know they can pack n pieces into n bins).

Table 1 shows the characteristic of the considered instances. The set of initial solutions are generated as follows:

Table 1: The one dimensional bin packing instances

Instances	Name	Capacity	No. pieces
1	Triples2004/instance1	1000	2004
2	Falkenauer/u1000-01	150	1000
3	Test/testdual7/binpack0	100	5000
4	50-90/instance1	150	2000
5	Test/testdual10/binpack0	100	5000

- Generate a random sequence of items
- Pack them one by one into the first bin into which they will fit first fit heuristic
- Evaluate the quality of solution using Eq. 2 (Hyde *et al*., 2010)

To evaluate the fitness of the solution researchers employ an alternative fitness function to the number of bins as in Eq. 2) (Hyde *et al*., 2010):

$$\text{Fitness} = 1 - \left(\frac{\sum_{i=1}^{n} \left(\frac{\text{fullness}_i}{C} \right)^2}{n} \right) \quad (2)$$

Where:

n = The number of bins
fullness_i = The sum of all the pieces in bin i
C = The bin capacity

This will avoid large plateaus in the search space around the best solutions.

Local search heuristics: In this research, the researchers implemented three basic local search heuristics. These are:

Hill Climbing/simple descent (HC): The heuristic choose the first improving neighbour that is better than the current solution. Then, an improving neighbour is immediately selected to replace the current solution. The neighbour is randomly evaluated. The heuristic terminate when there is no improved neighbours for certain iteration or when the termination criterion is met.

Simulated Annealing (SA): SA is a Stochastic Algorithm that probabilistically accepts some worse solutions to escape from local optimum. It begins by randomly or heuristically generates an initial solution (or the initial solution is given). Then the search starts with a high initial temperature to allow diversification at early search. For each iteration, the temperature is gradually reduced to restrict the acceptance of low quality solution (intensification) and will end when it reach zero or nearly zero temperature (or when termination criterion is met). In each iteration a neighbour of the current solution is randomly generated. The qualities of the two solutions (the current, $f(s_0)$ and neighbor solution, $f(s')$) are compared ($\delta = f(s') - f(s_0)$). A decision is made whether the new solution should be accepted. An improved solution is always accepted. However, in order to escape from the local optimum, a worse solution will probabilistically accepts that depends on SA acceptance criterion (where if the generated random number is $< e^{-\delta/T}$ and T is the current temperature) (Talbi, 2009).

Multistart-simulated annealing: It combines the advantages of SA and multi-start hill climbing strategies. When the temperature is high, the SA acceptance criteria tend to perform a drastic search in the solution space to seek the solutions and there is a high probability that SA may accept worse solutions and quickly trapped in bad local optimum. Thus, SA that aspires to obtain (near) global optima typically requires some form of diversification to escape from local optimality. Without such diversification, SA may be restricted in a small area of the solution space, losing the possibility of finding a better local optimum (that might be a global optimum). The multi-start hill climbing strategy can provide an effective way of avoiding the search being trapped in local optimum. Therefore, the combination of the advantages of SA and the multi-start hill climbing strategy in escaping from local optimality may provide the rationale for developing a Multi-start Simulated Annealing (MSA) heuristic. In this research, the researchers restart the SA temperature when it cannot find an improved solution for a certain number consecutive iterations. Researchers reset the temperature to the value that the SA started trapped in local optimum.

RESULTS AND DISCUSSION

The experiments were conducted using the five test instances of one dimensional bin packing problem currently available in the 2011 HyFlex Software. As stated in CHeSC rules, the execution time is used as stopping condition which is determined by the Benchmark Software provided by CHeSC organizer (Ochoa *et al*., 2012). Thus, researchers performed ten runs for each instances and terminate after ten min runs. For all tested approaches (HC, SA and MSA) researchers only use a simple pertubation heuristic that is a simple swap where researchers select two different pieces at random and swap them if there is a space and if it will produce an improvement in fitness (available in Hyflex Software). The parameter setup used in this experiment is shown in Table 2.

Researchers report the best (minimum), average and the worst (maximum) quality of solution obtained by HC,

Table 2: Parameter setup for HC, SA and MSA

Local search algorithm	Parameters
Hill Climbing (HC)	Termination condition = 10 min
Simulated Annealing (SA)	Initial temperature: 80000
	Cooling schedule rate = 0.9999
	Termination condition = 10 min
Multistart-Simulated Annealing (MSA)	Initial temperature: 80000
	Cooling schedule rate = 0.9999
	Un-improvement counter = 20
	Termination condition = 10 min

Table 3: Results of five instances bin packing problems tested on HC, SA and MSA (minimum, maximum and average quality of solutions out of 10 runs)

	HC			SA			MSA		
Instances	Minimum	Maximum	Average	Minimum	Maximum	Average	Minimum	Maximum	Average
1	**0.007**	**0.011**	**0.008**	0.064	0.068	0.064	0.046	0.057	0.051
2	**0.007**	**0.011**	**0.008**	0.059	0.064	0.062	0.049	0.049	0.049
3	**0.022**	**0.023**	**0.022**	0.029	0.030	0.029	0.026	0.028	0.027
4	**0.024**	**0.025**	**0.025**	0.033	0.033	0.033	0.031	0.031	0.031
5	**0.000**	**0.005**	**0.001**	0.022	0.022	0.022	0.015	0.020	0.018

Table 4: Results of five instances bin packing problem tested on HC, SA and MSA (best quality solutions out of 10 runs) comparing to the top five hyper-heuristics

	The tested heuristics			The top five hyper-heuristic framework from CHeSC competition				
Instances	HC	SA	MSA	AdapHH	VNS-TW	ML	Phunter	EPH
1	**0.007**	0.0640	0.0460	0.01310	0.0298	0.03230	0.0397	0.04300
2	0.007	0.0590	0.0490	**0.00280**	0.0036	0.00670	0.0034	0.00340
3	0.022	0.0290	0.0260	0.00040	0.0136	0.01240	0.0178	0.00800
4	**0.024**	0.0330	0.0310	0.10830	0.1087	0.10840	0.1088	**0.10830**
5	**0.000**	0.0220	0.0150	0.00310	0.0238	0.01780	0.0318	0.01360
Ave. overall	**0.012**	0.0414	0.0334	0.02554	0.0359	0.03552	0.0403	0.03526

Ave. overall: The overall average for instances. Bold values indicate the best quality values among them

SA and MSA out of ten runs (Table 3). Results demonstrate that in term of solution quality, HC out performed SA and MSA in all instances (where the best, average and the worst quality solution obtained by HC are better than SA and MSA).

To further evaluate the performance of HC, SA and MSA researchers compare these heuristics with the top five Hyper-Heuristic Methods from CheSC competition (AdapHH, VNS-TW, ML, PHUNTER and EPH) (Ochoa *et al*., 2012) based on the best quality solution (Table 4). The results demonstrate that out of 5, HC outperformed the top five Hyperheuristic Methods on 3 instances (1, 4 and 5 instance) and being inferior on 2 instances (2 and 3 instance). Researchers can also observe that HC manage to produce new best results for 3 instances. In average, HC is superior to the top five Hyper-Heuristic Methods from CheSC competition (based on Ave. Overall).

CONCLUSION

This research had implemented three basic local search heuristics (hill climbing, simulated annealing and multi-start simulated annealing) to solve the one dimensional bin packing problems. Researchers use the five test instances of one dimensional bin packing problem currently available in the 2011 HyFlex Software. Results indicated that hill climbing manage to find good quality solutions in a short run. This is due to its simplicity and perhaps at early stage there are many unexplored neighbours. Whilst other advanced heuristics are computational expensive (compared to hill climbing) and therefore spend more time to explore the search.

However, the advanced heuristics may perform well if the running is expanded. This shows that the intelligent mechanism in advanced metaheuristic approaches give negative effect to the performance of the metaheuristic at the early search time that is the hypothesis is not supported.

ACKNOWLEDGEMENT

Researchers wish to thank Ministry of Higher Education for supporting this research under the FRGS Research Grant Scheme (FRGS/1/2012/SG05/UKM/02/11).

REFERENCES

Blum, C., M.J.B. Aguilera, A. Roli and M. Samples, 2008. Hybrid Metaheuristics: An Emerging Approach to Optimization. Springer, USA., pp: 1-30.

Cook, S.A., 1971. The complexity of theorem-proving procedures. Proceedings of the 3rd annual ACM symposium on Theory of computing, May 3-5, 1971, New York, USA., pp: 151-158.

Dorigo, M. and L.M. Gambardella, 1997. Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Trans. Evol. Comput., 1: 53-66.

Dowsland, K.A., E. Soubeiga and E. Burke, 2007. A simulated annealing based hyperheuristic for determining shipper sizes for storage and transportation. Eur. J. Oper. Res., 179: 759-774.

Hyde, M., G. Ochoa, J.A. Vazquez-Rodriguez and T. Curtois, 2010. A hyflex module for the one dimensional bin packing problem. School of Computer Science, University of Nottingham, Technical Reports.

Martello, S. and P. Toth, 1990. Knapsack Problems: Algorithms and Computer Implementation. John Wiley and Sons, New York, ISBN: 0-471-92420-2, pp: 296.

Ochoa, G., M. Hyde, T. Curtois, J.A. Vazquez-Rodriguez and J. Walker *et al*., 2012. HyFlex: A Benchmark framework for cross-domain heuristic search. Proceedings of the 12th European conference on Evolutionary Computation in Combinatorial Optimization, April 11-13, 2012, Malaga, Spain, pp: 136-147.

Reeves, C.R. and J.E. Rowe, 2002. Genetic Algorithm-Principles and Perspectives: A Guide to GA Theory. Kluwer Academic Publishers, Boston, USA.

Schreiber, G.R. and O.C. Martin, 1999. Cut size statistics of graph bisection heuristics. SIAM J. Optim., 10: 231-251.

Talbi, E.G., 2009. Metaheuristics: From Design to Implementation. John Wiley and Sons, Hoboken, New Jersey.

Wolpert, D.H. and W.G. Macready, 1997. No free lunch theorems for optimization. IEEE Trans. Evol. Comput., 1: 67-82.