



BOOSTING SIMULATED ANNEALING WITH FITNESS LANDSCAPE PARAMETERS FOR BETTER OPTIMALITY

Sunanda Gupta, Sakshi Arora

Shri Mata Vaishno Devi University, Kakryal, Katra, Jammu and Kashmir, India
 e-mail: sunanda.gupta@smvdu.ac.in, sakshi@smvdu.ac.in, smvdu.net.in

Abstract: Multi Dimensional Knapsack problem is a widely studied NP hard problem requiring extensive processing to achieve optimality. Simulated Annealing (SA) unlike other is capable of providing fast solutions but at the cost of solution quality. This paper focuses on making SA robust in terms of solution quality while assuring faster convergence by incorporating effective fitness landscape parameters. For this it proposes to modify the 'Acceptance Probability' function of SA. The fitness landscape evaluation strategies are embedded to Acceptance Probability Function to identify the exploitation and exploration of the search space and analyze the behavior on the performance of SA. The basis of doing so is that SA in the process of reaching optimality ignores the association between the search space and fitness space and focuses only on the comparison of current solution with optimal solution on the basis of temperature settings at that point. The idea is implemented in two different ways i.e. by making use of Fitness Distance Correlation and Auto Correlation functions. The experiments are conducted to evaluate the resulting SA on the range of MKP instances available in the OR library. *Copyright © Research Institute for Intelligent Computer Systems, 2015. All rights reserved.*

Keywords: Optimization, Multi Dimensional Knapsack Problem.

1. INTRODUCTION

Multidimensional Knapsack Problem (MKP) is an NP hard combinatorial optimization problem [1]. The problem is stated as follow: a set of m knapsack constraints and n items is given. For each item j (when $1 \leq j \leq n$), a profit P_j is incurred, and a resource consumption r_{ij} is given for each constraint i where $1 \leq i \leq m$. The objective is to identify a set of items that maximizes the total profit keeping in view that the given constraint capacity c_i does not get violated. MKP is defined as follows:

$$\text{maximize } \sum_{j=1}^n P_j x_j \quad (1)$$

$$\text{subject to } \sum_{j=1}^n r_{ij} x_j \quad \forall i = 1 \dots m \quad (2)$$

$$x_j \in \{0,1\}, j = 1 \dots n$$

$$\text{with } P_j \geq 0, r_{ij} \geq 0 \text{ and } c_i \geq 0$$

Several heuristic and Meta heuristic approaches have been applied in solving MKP like Genetic Algorithm (Chu and Beasley [2]), Binary Trie Coding Scheme (Sunanda and M.L. Garg [3]), Tabu

Search (Pirkul [4]), Global Swarm Optimization (Hembeckler, Lopes and Godoy [5]), Simulated Annealing (Qian and Ding [6]).

Simulated Annealing a probabilistic local search algorithm was proposed in the year 1983 by Kirkpatrick, Gelatt and Vecchi [7]. The strength of SA lies in its ability to get executed easily and converge faster to local optima.

In this paper the performance of SA is examined by analyzing the effect of fitness landscape analysis techniques on the quality of solutions generated by the binary encoded SA. The two fitness landscape analysis techniques being augmented to SA are Fitness Distance Correlation (FDC) and Auto Correlation Function (AC) [9]. To be precise the questions that need to be explored and addressed are as follows:

- Does the Self adaptive Acceptance Probability function based on the outcome of 'landscape analysis' technique work well for the binary encoded SA? i.e. Does it outperform the standard SA.
- Analytically examining the performance by changing parameters like temperature selectively or iteratively.

The paper is structured as follows: In section 2 we describe the working of Simulated Annealing (SA) and the fitness landscape analysis techniques

FDC and AC in conjunction to it, for developing the Acceptance probability function. Section 3 proposes the modified Acceptance probability function. Section 3 describes the benchmark instances of MKP used in our experiments, the performance parameters measured for comparing standard SA with other variants and the experimental results. Section 5 provides the conclusion.

2. RELATED WORK

Simulated Annealing avoids getting entrapped in local optima by injecting the appropriate amount of randomness to it. In the process of getting global optima it accepts to keep worse solutions at times, based on the ‘acceptance probability’ and then compare to a random number. The equation typically used for the acceptance probability is

$$\alpha = e^{\frac{f_{new}-f_{old}}{T}} \quad (3)$$

Here T represents the temperature with initial value of 1 which is decreased at the end of each iteration by multiplying it by a constant called α , the temperature coefficient. SA does better when around 100 to 1000 iterations are performed at each temperature. The generic flowchart of Standard SA is shown below.

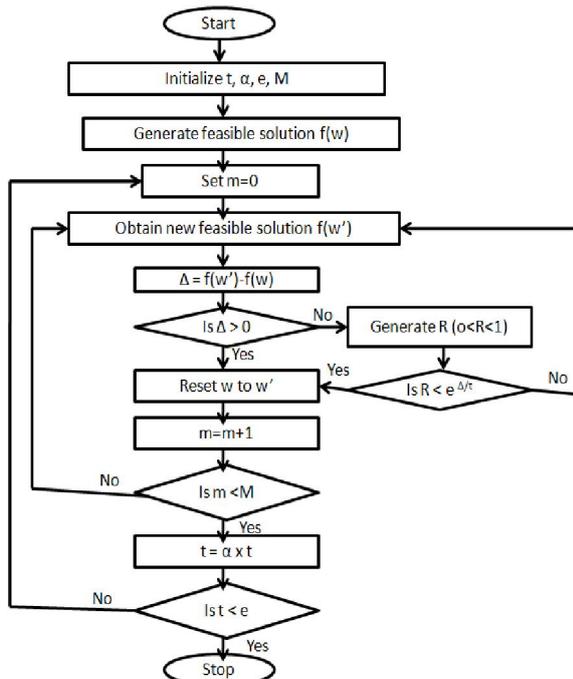


Fig. 1 – Generic Flow of Standard Simulated Annealing Approach.

Literature reveals that SA is an efficient technique to discover good quality solutions fast for combinatorial optimization problem like MKP [8].

But when we compare its performance with other meta heuristics we observe that it reaches optimality much faster and easier in comparison to them but with a compromise on the quality of solution.

We also find from literature a strong association between the search space, the fitness space and the phenotypic space that defines the fitness landscape [9] of an evolutionary algorithm. An evolutionary algorithm in the process of finding the optima traverses through the landscape and uses the landscape analysis techniques as tools in determining how EA works. Embedding landscape analysis techniques to acceptance probability function of SA can be used as a decisive parameter in selecting or rejecting a new solution of inferior quality. The already existing landscape analysis techniques being used here are Fitness Distance Correlation (FDC) and Autocorrelation Function (ACF) [9].

2.1 FITNESS DISTANCE CORRELATION (FDC)

FDC determines how strongly the fitness value and the distance to the nearest optimal solution are related. It is computed as follows [9]:

$$\rho(f, d) \approx \frac{1}{\sigma_f \sigma_{d_m}} \sum_{i=1}^m (f_i - f')(d_i - d') \quad (4)$$

Here, $f_i = f(x_i)$ represents the fitness value and $d_i = d_{opt}(x_i)$ represent the minimum distance to the global optima. f', d' are the mean values of fitness and distance whereas σ_f and σ_{d_m} represent the standard deviation respectively. Higher fitness values with small distance to optima represent easy search and a value of -1 represents a perfect relation of fitness and distance indicating a strong correlation. A value of 1.0 indicates opposite.

2.2 AUTO CORRELATION FUNCTION (AC)

AC examines the characteristics of landscape by evaluating the degree of correlation between various points of the landscape which is calculated on the basis of difference between their fitness values [9]. Smooth landscapes represent strong correlation making the search easier whereas higher differences among the fitness values depicts less correlated rugged landscape, thus making the search more difficult.

$$\rho_{rw}(s) \approx \frac{1}{\sigma_f^2(m-s)} \sum_{t=1}^{m-s} (f(x_t) - f')(f(x_{t+s}) - f') \quad (5)$$

In this case a random walk $f(x_t)$ defines the correlation of two points s steps away in a m long random walk [9].

3. PROPOSED STRATEGY

We know from the SA approach that with higher temperature, the probability of selecting an inferior move is high, thus encouraging candidate solutions to explore more regions of solution space at early stages of exploration. The objective is to keep the solution from converging to local optimum.

This forms the basis of incorporating effective landscape analysis techniques for adjusting the temperature settings instead of following a decremental approach and accordingly adjusting the random function in designing an effective acceptance probability function. This function would employ an adaptive mechanism that tries to create a balance between the exploration and exploitation properties in the convergence process of standard SA. The balance between these parameters of SA is created by regulating the values of temperature and random function using the behavior of fitness landscape. The brief outline of Acceptance probability function as given in [6] is as follows:

$$P\{\text{Accept } w' \text{ as next solution}\} = \begin{cases} \exp\left[\frac{f(w') - f(w)}{t_k}\right] & \text{if } f(w') - f(w) < 0 \\ 1 & \text{if } f(w') - f(w) \geq 0 \end{cases} \quad (6)$$

The new computed solution w' is accepted with the acceptance probability calculated above and on the basis of random function. $R = \text{random number } (0 < R < 1)$. If the new computed solution is better than old it is accepted as it is otherwise it is accepted if $R < P$.

Algorithm 1 represents the Acceptance Probability function embedding FDC, and Algorithm 2 represents Acceptance Probability Function embedding AC. The function $FDC()$ in algorithm 1 is calculated using equation (4) and the function $AC()$ in algorithm 2 is computed using equation (5).

Algorithm 1: Fitness Distance Correlation Based Acceptance Probability Function

FDC Acceptance Probability Function ($\Delta\text{change}, T$)
 if (Δchange) > 0
 accept move
 else $\Gamma = FDC()$
 if (Γ) = -1 then
 {
 accept move
 $t_k = t_k \times \alpha$
 and reinitialize $m = M$
 }
 else { $R = \text{random } (0 < R < 1)$

$Y = \exp\left[\frac{f(w') - f(w)}{t_k}\right]$
 if ($R < Y$) then accept move
 else Reject move
 end

Algorithm 2: Auto Correlation Based Acceptance Probability Function

AC Acceptance Probability Function ($\Delta\text{change}, T$)
 if (Δchange) > 0
 accept move
 else $\rho = AC()$
 $\ell = \frac{-1}{\ln |\rho(1)|}$
 if $\ell \geq 0.7$ and $\ell \leq 1$ then
 {
 accept move
 $t_k = t_k \times \alpha$
 and reinitialize $m = M$
 }
 else { $R = \text{random } (0 < R < 1)$ skewed with probability ℓ
 $Y = \exp\left[\frac{f(w') - f(w)}{t_k}\right]$
 if ($R < Y$) then accept move
 else Reject move
 end

In a normal scenario after each iteration temperature t is scaled by a cooling factor α where $0 \leq \alpha \leq 1$ but in the case of above two functions the temperature t is immediately decremented by a cooling α and m reset to M . The reason attributed to this is that under normal circumstances SA does better when the neighbor cost compare moves is carried out multiple times; but by incorporating efficient landscape analysis tools we can judge the quality of solutions generated viz-a-viz the fitness landscape i.e. how well the exploitation of the search space has taken over the given temperature settings and when justified resetting m to M for that temperature indicating the best found solution.

In algorithm 2, ℓ is a measure of correlation length that quantifies the ruggedness of the landscape. It is computed on the basis of formula defined above which is basically dependent on the Auto Correlation Function [9] given in equation (4). The closer is correlation length to 1, higher is the probability of smother landscape and farther it is from, higher is the probability of rugged landscape. This parameter is decisive in controlling the behavior of random function.

4. EXPERIMENTATION AND RESULTS

This paper focuses on implementing finite time implementation of SA approach for MKP in order to find fairly good quality solutions in optimal time. The test instances of MKP have been taken from

Beasley’s OR Library [10]. We have assumed t_0 the initial temperature to be 500 by rough estimation of test instances. Here, t_0 depicts the maximum permitted variation between two neighboring solutions. Final temperature e is taken to be $1.0e^{-5}$ as given in [6]. The generic flow that SA follows for MKP is taken from [6] given by Qian Fubin and Ding Rui. Fig. 2 shows solution quality and Fig. 3 represents the CPU time by varying Markov chain length M ranging from 10 to 200 and for $\alpha = 0.85$ using the *FDC Acceptance Probability Function()* and *AC Acceptance Probability Function()* and compares with the results given by Qian Fubin and Ding Rui [6].



Fig. 2 – Solution Quality of SA using Qian and Dings Probabilistic function [6], FDC Acceptance Probability function and AC Acceptance Probability Function with different Markov Chain Length.

From the results obtained from Fig. 2, the following inferences were drawn: The desirable length for the Markov Chain yielding optimal results for the Qian and Dings approach [6] and SA using *FDC Acceptance Probability Function()* is approximately between 13 to 14, whereas for SA incorporating *AC Acceptance Probability Function()*, the desirable length is around 9. It is also inferred that the performance of SA in conjunction with *AC Acceptance Probability Function()* or *FDC Acceptance Probability Function()* does not affect the performance for smaller values of M . However, a significant improvement in quality of solutions is achieved with *AC Acceptance Probability Function()*. The *AC Acceptance Probability Function()* exhibits better performance in comparison to *FDC Acceptance Probability Function()*.

By comparing the results from Fig. 3, it was inferred that the *AC Acceptance Probability Function()* or *FDC Acceptance Probability Function()* require more time for the instances when M is less than equal to 4. However for rest of the instances FDC takes least time in comparison to the other two approaches. The reason attributed to more computation time of *AC Acceptance Probability*

Function() in comparison to *Fitness Distance Correlation Function()* reflects the robustness of AC in evaluating the ruggedness or smoothness of the landscape. In totality the performance of SA incorporating *AC Acceptance Probability Function()* is considered best as it yields the best solution for the case when the Markov chain length M is approximately around 9. Hence Auto Correlation measure when incorporated with Acceptance Probability function, promotes both diversity and fitness.

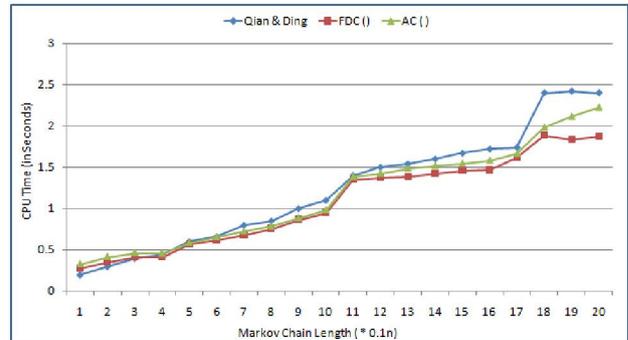


Fig. 3 – Summary of CPU time of SA using Qian and Dings Probabilistic function [6], FDC Acceptance Probability function and AC Acceptance Probability Function with different Markov Chain length.

Fig. 4 provides an insight to the performance of SA by varying the temperature control parameter α , ranging from 0.8 to 1.0. It can be inferred that SA using *FDC Acceptance Probability Function()* and *AC Acceptance Probability Function()* yields optimal results when α is around 0.85.

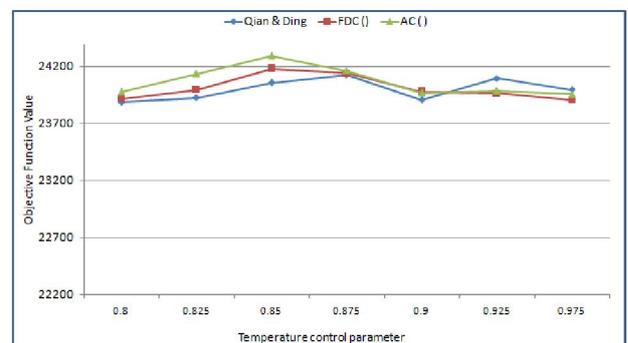


Fig. 4 – Solution Quality of SA using Qian and Dings Probabilistic function [6], FDC Acceptance Probability function and AC Acceptance Probability Function with different values of α .

Results of Simulated Annealing featuring AC Acceptance probability function are further compared with Binary Trie Coding Scheme (BTCS)[3], an enhanced hybrid genetic algorithm used for solving 0/1 Multi dimensional knapsack

problem. The robustness of BTCS lies in its ability to provide better coverage of the search space by assuring uniqueness of individuals during the process of evolution and imposing stringent measures to identify the saturation of sub spaces within the search space. The below given table gives the percentage gaps ($100 \times (\text{optimumLP} - \text{optimumGA})/\text{optimumLP}$) relative to the solution of LP relaxation available in the OR library [10].

Table 1 compares the performance of SA-AC with other heuristics available in literature. It illustrates the percentage gaps for the instances, when n is equal to 500 by varying m to 5, 10 and 30. A represents the tightness ratio.

GBAM+ denotes the penalty based technique that uses adaptive mutation scheme for fast convergence to optimum value [11], GA denotes the repair based EA proposed by Chu and Beasley [2], BTCS is an enhanced genetic algorithm which avoids inclusion of duplicates during evolution [3], and SA-AC is the proposed approach incorporating AC Acceptance Probability function. The percentage gap values for GBAM+, GA and BTCS are taken from the reported values in the corresponding tables given in [11], [2] and [3].

Table 1. Percentage Gap values for GBAM+, GA, BTCS and SA-AC for the case n = 500.

n	m	α	GBAM+	GA	BTCS	SA-AC
500	5	0.25	0.59	0.09	0.0933	0.29
		0.5	0.21	0.04	0.0443	0.15
		0.75	0.09	0.03	0.0429	0.56
	10	0.25	0.6	0.24	0.2424	0.42
		0.5	0.27	0.11	0.0946	0.16
		0.75	0.15	0.07	0.0501	0.09
	30	0.25	0.97	0.61	0.6105	0.78
		0.5	0.43	0.26	0.2387	0.35
		0.75	0.28	0.17	0.1738	0.32

The significance of evaluating Percentage gaps is to identify how optimal the heuristic solutions are i.e. determining how near they are to the linear programming solution. As is evident from the results BTCS outperforms other heuristic approaches primarily because of the Virtually compressed binary trie structure [3] that ensures uniqueness of individuals as well as its ability to identify convergence.

GBAM+ makes use of penalty based technique and literature reveals that its results are superior to many of the existing penalty based techniques [11]. But, when the percentage gap values of GBAM+ are compared with BTCS, GA or SA-AC, it lags behind. The results of BTCS are superior as it works only on boundary solutions. SA-AC provides an improvement over solutions provided by GBAM+, signifying the importance of embedding the AC

acceptance probability function to SA, ensuring knowledge based searching mechanism for faster and effective convergence.

Table 2 further provides the comparison of the mentioned heuristics [2] and [3] with SA-AC in terms of computational cost. It provides the average best solution time for the case when n equals 500. Average Best Solution Time refers to the time taken to achieve the optimal solution and is always less than the average execution time. Average execution time is the time dependent on the completion of the process. GA gives the average best solution time as given in [2]. BTCS gives the average best solution time (ABST) as given in [3] and SA-AC is the average best solution time computed by the proposed approach. ABST of GA is the highest due to its inability to deal with recurrence of individuals during evolution which in a way delays the convergence process. ABST of BTCS is lesser than that of GA but still high due to the Virtually compressed binary trie structure being updated with every inclusion of the unique individual and also due to its ability in tracking duplicates. SA-AC when compared with these two provides solutions much faster because of the simplicity with which SA works.

Table 2. Average Best Solution Time of GA, BTCS and SA-AC for the case n = 500.

n	m	α	GA	BTCS	SA-AC
500	5	0.25	264.6	147.8	32.35
		0.5	291.3	162.7	59.89
		0.75	386.2	215.8	102.3
	10	0.25	702.5	392.5	126.32
		0.5	562.2	314.1	128.2
		0.75	937.6	523.8	183.5
	30	0.25	1127.2	629.7	232.8
		0.5	1121.6	626.6	257.2
		0.75	1903.2	1063.1	301.2

5. CONCLUSION

The objective of this study was to explore fitness landscape analysis techniques with SA, to better understand the correlation between SA control parameters (Markov chain length m, temperature control parameter α) on the SA performance viz-a-viz the two incorporated approaches.

From the results it was inferred that SA with *AC Acceptance Probability Function()* outperforms both Qian and Dings approach [6] and SA using *FDC Acceptance Probability Function()*, both in terms of solution quality and computational time. Computational time, because it provides better quality solutions for smaller values of M, than in comparison to the other two approaches (which fail to produce match able solutions even for larger values of M). It is also inferred that the best possible

combination of SA parameters; the Markov chain length, M is 9 (approx.) and temperature control parameter α is 0.85 to achieve optimality.

Further, on comparison of SA using AC Acceptance probability function with heuristics BTCS [3], GBAM+ and GA [2] it was inferred that though the solution quality of BTCS and GA is better than that of SA-AC, but at the cost of computational time. SA-AC provides a good tradeoff in terms of computational time and solution quality when compared with other heuristics.

6. REFERENCES

- [1] S. Martello and Paolo Toth, *Knapsack problems: algorithms and computer implementations*, John Wiley & Sons, Inc., 1990.
- [2] P.C. Chu, and John E. Beasley, A genetic algorithm for the multidimensional knapsack problem, *Journal of Heuristics*, (4) 1 (1998), pp. 63-86.
- [3] S. Gupta, and M.L. Garg, Binary trie coding scheme: an intelligent genetic algorithm avoiding premature convergence, *International Journal of Computer Mathematics*, (90) 5 (2013), pp. 881-902.
- [4] H. Pirkul, A heuristic solution procedure for the multiconstraint zero/ one knapsack problem, *Naval Research Logistics*, (34) 2 (1987), pp. 161-172.
- [5] F. Hembeker, H.S. Lopes, and W. Godoy Jr., Particle swarm optimization for the multidimensional knapsack problem, *Adaptive and Natural Computing Algorithms*, Springer Berlin Heidelberg, 2007, pp. 358-365.
- [6] F. Qian, and R. Ding, Simulated annealing for the 0/1 multidimensional knapsack problem, *Numerical Mathematics – English Series*, (16) 4 (2007), pp. 320.
- [7] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, Optimization by simulated annealing, *Science*, (220) 4598 (1983), pp. 671-680.
- [8] W.L. Goffe, G.D. Ferrier and J. Rogers, Global optimization of statistical functions with simulated annealing, *Journal of Econometrics*, (60) 1 (1994), pp. 65-99.
- [9] J. Tavares, F. Baptista Pereira and E. Costa, Multidimensional knapsack problem: A fitness landscape analysis, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, (38) 3 (2008), pp. 604-616.
- [10] J.E. Beasley, OR-Library: distributing test problems by electronic mail, *Journal of the*

Operational Research Society, (1990), pp. 1069-1072.

- [11] S. Uyar and G. Eryiğit, Improvements to penalty-based evolutionary algorithms for the multi-dimensional knapsack problem using a gene-based adaptive mutation approach, in *Proceedings of the Conference on Genetic and Evolutionary Computation*, (2005), pp. 1257-1264.



Dr. Sunanda Gupta, She has Bachelors degree in Sciences and Masters degree in Computer Applications from University of Jammu, and PhD in Computer Science Engineering from School of Computer Science and Engineering, Shri Mata Vaishno, Devi University, India.

She is currently a faculty in School of Computer Science & Engineering, Shri Mata Vaishno Devi University, J&K, India, with more than ten years of teaching experience. She has authored several research articles in International journals of repute. Besides having presented papers in several international/ national conferences she has been invited as an expert to various international conferences as paper reviewer/ program technical committee member. Her research interests include combinatorial optimization problems, genetic algorithms and image processing.



Dr. Sakshi Arora, She has Bachelors degree in Sciences and Masters degree in Computer Applications from University of Jammu, and PhD in Computer Science Engineering from School of Computer Science and Engineering, Shri Mata Vaishno Devi University, India.

She is currently a faculty in School of Computer Science & Engineering, Shri Mata Vaishno Devi University, J&K, India, with more than ten years of teaching experience. She has published several research articles in various International journals. She has presented papers in several international/ national conferences; She has been associated with various international conferences as paper referee/ program technical committee member. Her research interests include cloud computing and virtualization, mobile computing and manets and met heuristic techniques for real world optimization.