New Discrete Metaheuristic Approach for Large Scale Problem

E. Songkroh and M. Anantathanavit

***Abstract*—Metaheuristic approaches have been widely used to solve large scale, complex global optimization problems. In this paper, a Hybrid simulated annealing based on discrete radius particle swarm optimization (H-DRPSOSA) with adaptive mutation is proposed. The proposed algorithm takes the advantage of the global search of the RPSO and the local search strategy of the SA algorithm to quickly generate good solutions.The paper also explains the framework design to solve the large scale multidimensional knapsack problems (LCO-MKPs). Additional, we present a random transfer mechanism for the feasible solution of the discrete search region. The proposed hybrid is compared to state-of-the-art solution techniques by applying them to the multidimensional knapsack dataset. Computational results demonstrate that the proposed algorithm is capable of producing competitive solutions.**

***Index Terms*—Large Scale Complex Problems (LCOs), Radius Particle Swarm Optimization (RPSO), Simulated Annealing (SA), Multidimensional Knapsack Problems (MKPs)**

I. Introduction

Large scale complex problems (LCOs) explore a large number of the feasible solution in the search space such as resource constrained project scheduling problem (RCPSP), Traveling salesman problem (TSP) and knapsack problems(KPs) [1]. LCOs can be divided into several categories such as continuous or discrete and constrained or unconstrained [2]. In the encounter of increasing complexity and dimensionality of LCOs, some investigations [3] are adapted to type of LCOs. The classical metaheuristic algorithms were usually designed for low dimensional problems and show the low performance during tackling high-dimensional problems. Many modified algorithms [4], [5] have been proposed strategies on the classical technique and applied on different stages to enhance the performance of algorithms.

The large scale multidimensional knapsack problems (LS-MKPs) can be modeled in the discrete domain with constrained value and variety of practical problems, such as capital budgeting controlling, resources allocation [6]-[8], web polling problem [9], project selection [10], cutting and packing problem [11], and cryptography [12]. The MKPs is given a set of items with weights and sizes, and the capacity value of a knapsack, to maximize the total weight of selected items in the knapsack satisfying the capacity constraint [13].The MKPs have been studied in the last few decades, involving both exact and heuristic algorithms [14]-[16]. Bansal and Deep [17] propose the modified binary particle swarm optimization (MBPSO) that updates the term of position of BPSO gives a new probability of selection. The amoeboid organism model is proposed by Zhang *et al*. [18]. They convert the problem to a directed domain by the network converting algorithm. Wan *et al* [19] is developed the new probability model based on estimation of distribution algorithm (EDAs) on the candidate solutions.

In this paper, we present a hybrid algorithm of discrete radius particle swarm optimization (DRPSO) with adaptive mutation and simulated annealing (SA) to solve LCO-MKPs using the maximum profit as the objective function. Additional, we present a random transfer mechanism for the feasible solution. Thus, the data used for the experiments in this paper has been derived from the standard dataset of multidimensional knapsack problems (MKPs).

The multidimensional knapsack problems (MKPs) is introduced in Section II. Section III describes the discrete radius particle swarm algorithm (DRPSO), simulated annealing (SA) and framework of H-DRPSOSA for MKPs**.** Then, the experiments and results are illustrated and analyzed in Section IV. Finally, Section V provides conclusion.

II. Multidimensional Knapsack Problem

The multidimensional knapsack problems (MKPs) is a type of LCOs. The MKPs is a typical NP-hard problem in operations research. Thus, the MKPs is given a set of items with weights and values. The capacity value of knapsack to maximize overall profit of selected items whose total weight does not exceeding the capacity of knapsack. The MKPs can be given as equation (1)

$$\text{Maximize } f(x) = \sum_{i=1}^{n} p_i\, x_i \quad (1)$$

Subject to

$$\sum_{j=1}^{n} w_{i,j} x_i \le C_j \quad \forall j = 1,2,3,\dots,m, w_{i,j} \ge 0, \; C_j \ge 0, \; x_i \in \{0,1\}, i = 1,2,3,\dots,n$$

where n is the number of items, m is the number of knapsack constraints with capacities C_j $(j = 1,2,3,\dots,m)$ associated weights constrains matrix $w_{i,j}$, and p_i is the positive profit of the item i . x_i is a binary variable$\{0,1\}$ that indicates $x_i = 1$, if the item i has been entered in the knapsack and $x_i = 0$, if the

Manuscript received March 7, 2021; revised December 22, 2021. This work was supported by Rajamangala University of Technology Suvarnabhumi, Thailand.

E. Songkroh is with Rajamangala University of Technology Suvarnabhumi, Thailand (e-mail: ekawit.s@rmutsb.ac.th).

M. Anantathanavit was with Mahanakorn University of Technology, Thailand (e-mail: m.anantathanavit@gmail.com).

doi: 10.18178/ijmlc.2022.12.4.1090

knapsack remain out.

III. Related Work

The swarm intelligence (SI) are known to deal with the optimization problem. SI is the kinds of heuristic algorithm, called metaheuristic algorithm. SI have a concern with collective intelligence of social insects such as ants, bees and birds. SI are based on the collective behavior and self-organization of population. The behavior and self-organization in nature is called colony. The inspection of behavior of the colony dynamically and adaptively collect optimal solution from the search space. These action may change the environment and its neighbors by its knowledge. The idea of process is related the information about a promise position to obtain the candidate solution.The performance of SI depends on a balance between the exploitation and exploration on working process.

Regarding swarm intelligence, on which we will focus in this the paper, is the Radius Particle Swarm Optimization (RPSO), brief overview of RPSO is descripted in the next section to provide a background for the proposed algorithm.

A. Radius Particle Swarm Optimization with Adaptive Mutation

Particle Swarm Optimization (PSO) [20] is a swarm intelligence algorithm that operates via a simple technique inspired by the flocking behaviour of birds and is referred to as a metaheuristic algorithm. The particle flies to a new position according to its new velocity and previous positions. Due to the fast convergence and easy implementation, PSO suffers from the premature convergence problem when dealing with complex optimization problems because it is easy to be trapped into local optima. Thus, the trapped particles will not participate in the search space.

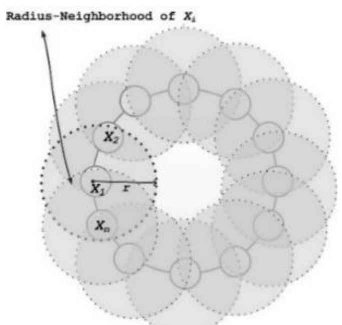

Fig. 1. The swarm circle topology and a radius-neighbourhood.

The Radius Particle Swarm Optimization (RPSO) was proposed in [21] as the efficient algorithm for the complex optimization problem in which the scale factor is correlated with the numerical benchmarks problem. Thus, the RPSO extend the traditional PSO algorithm by grouping particles within the same radius into a new particle agent and then iteratively finding the best solution under the given objective functions. The significant concept of the RPSO is based on the lbest circle topology finding the agent particle within the radius of a circle, as shown in Fig. 1. Thus, it uses a swarm circle topology to find the agent particle within the radius of the circle. Each particle in the overlap radius can be in multiple groups. Once a group is defined, we find the best particle in that swarm group and assign to the agent particle represented by $abest_{i,j}$, as shown in Fig. 2. Finally, the agent particles are the candidates for finding the optimal solution, or the gbest position, as shown in Fig. 3. To overcome the premature convergence problem, the RPSO takes advantage of group-swarms to maintain the swarm diversity and evolution by sharing information from the agent particles, which effectively keep the balance between the global exploration and the local exploitation. Obviously, the agent particle guides the neighbouring particles to jump out of the local optimum and achieve the global best.

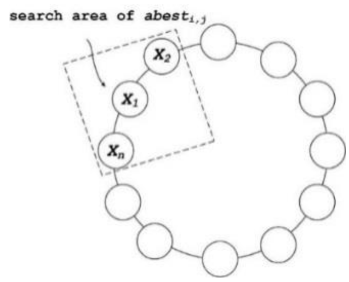

Fig. 2. The radius-neighbourhood for the agent particle (abest).

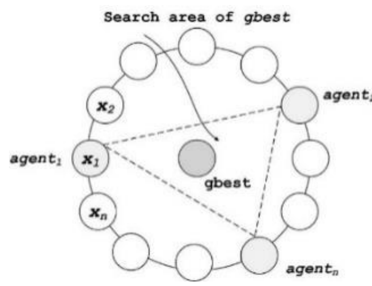

Fig. 3. The global neighbourhood for the global best (gbest).

The euclidean distance is used to calculate the radius-neighbourhood between particle i and particle θ using equation (2).

$$d(p_i, p_\emptyset) = \sqrt{\sum_{j=1}^{m}\left(p_{i,j} - p_{\emptyset,j}\right)^2} \;; d \leq 2r\,,$$

$$i \neq \emptyset\,,\ for\ \ 1 \leq\ i\ \leq n\,, for\ \ \ 1 \leq\ \emptyset\ \leq n \tag{2}$$

where

$$\rho = \{x_{1,}x_2, x_3, \dots, x_n\}$$

$$r = \ \mu\ \cdot v_{max}; \mu \in [0.0,1.0] \tag{3}$$

In equation (3), the radius value (r) of the particle is obtained. Here, r is determined by the maximum velocity v_{max}. Therefore, v_{max} is assigned to the maximum bounds of the search space or the feasible bounds in the benchmark function. We consider the problem of finding the global optimum using the agent particle $(abest_{i,j})$ within a radius-neighbourhood as given in equation (4).

$$abest_{i,j} = \min_{\beta\in\rho} f(\beta) \tag{4}$$

Therefore, the particle i in the swarm updates it velocity and position as given in equation (5) and equation (6), respectively.

$$v_{i,j}(t+1) = w \cdot v_{i,j}(t) + c_1 \cdot R_1 \cdot \left(pbest_{i,j} - x_{i,j}(t)\right)$$

$$+c_2 \cdot R_2 \cdot \left(abest_{i,j}(t) - x_{i,j}(t)\right) \tag{5}$$

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1) \quad (6)$$

A drawback observed on the search space with the particles is the trapped on the the bounds of the search area (v_{max} or v_{min}) (See in Fig. 4). Thus, the trajectory of particle will be produced low exploration.

This paper present the adaptive mutation in RPSO with property of large exploration capability even in the case of large or low velocity values (v_{max} or v_{min}) to solve this problem.

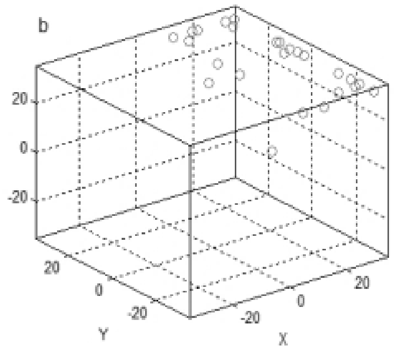

Fig. 4. The trajectory of PSO is trapped on the the bounds of the search area (v_{max} or v_{min}) of the search space.

The pseudo-code of the adaptive mutation algorithm is given below.

Algorithm 1. Adaptive mutation

if $x_{i,j}(t+1) > v_{max}$ **then**
    $\gamma = random(1, i)$
    $x'_{i,j}(t) \leftarrow x_{\gamma,j}(t)$
    $x_{i,j}(t+1) \leftarrow x'_{i,j}(t)$
end if
if $x_{i,j}(t+1) < v_{min}$ **then**
    $\gamma = random(1, i)$
    $x'_{i,j}(t) \leftarrow x_{\gamma,j}(t)$
    $x_{i,j}(t+1) \leftarrow x'_{i,j}(t)$
end if

As limited by the bounds of the search area, the mutation operator is applied to each dimension of the particle. The adaptive mutation procedure is elaborated as follows:

Step 1. Random number γ between 1 and the number of particle i.

Step 2. Set the particle $x'_{i,j}(t)$ for the mutation from the particle $x_{\gamma,j}(t)$ of number γ in the swarm.

Step 3. Select the mutation points δ_j on the selected particle $x_{i,j}(t+1)$. For each mutation point δ_j, the $x_{i,j}(t+1)$ is inserted into the new position $x'_{i,j}(t)$.

The pseudo-code of the RPSO with adaptive mutation algorithm is given below.

Algorithm 2. RPSO algorithm with Adaptive mutation

for each time step t to t_{max} **do**
    for each particle i in the swarm **do**
        update position $x_{i,j}(t+1)$ with Eq.(5) and Eq.(6)
      if $x_{i,j}(t+1) > v_{max}$ **then Adaptive mutation**
        if $x_{i,j}(t+1) < v_{min}$ **then Adaptive mutation**
        calculated particle's fitness $f(x_{i,j}(t+1))$
       if particle's fitness $f(x_{i,j}(t+1))$ **better than** $pbest\ x_{i,j}(t+1)$ **then update** $pbest\ x_{i,j}(t+1)$
        find $agent$ particle
       if agent particle's fitness $f(abest_{i,j}(t+1))$ **better than** gbest
          then update $gbest$
    end for
end for

B. Discrete Radius Particle Swarm Optimization

The MKPs require algorithms that can operate in discrete search space. We present the Discrete Radius Particle Swarm Optimization (DRPSO) for the MKPs. Each particle takes the value of one or zero for its position within the probability. The updated position is redefined by the role (equation 7) and updates the position by equation (8). In this paper, we apply the concept of DRPSO algorithms to solve the MKPs.

$$S\left(v_{i,j}(t+1)\right) = \frac{1}{1+e^{-v_{i,j}(t+1)}} \quad (7)$$

$$x_{i,j}(t+1) = \begin{cases} 0 & if\ rand() \geq S\left(v_{i,j}(t+1)\right) \\ 1 & if\ rand() < S\left(v_{i,j}(t+1)\right) \end{cases} \quad (8)$$

where $rand()$ is the random number between 0 and 1, $S()$ in the equation 7 is the sigmoid function (see in Fig. 5) for transforming the velocity to the probability value.

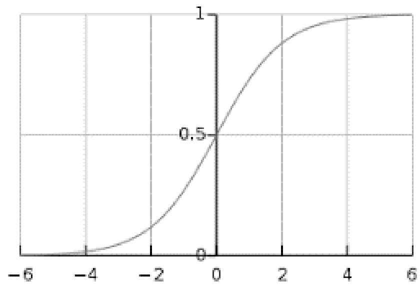

Fig. 5. The Sigmoid function.

C. Simulated Annealing

Kirkpartick *et al*. [22] proposed simulated annealing (SA) algorithm that is a gradient method for the optimization problem. The SA gradually improves the candidate solution by searching for optimal solution within a local neighborhood. There are two ways of accepting a new solution. First, if its fitness value is better than that of the current solution. Second, in worse fitness case accepts a solution with a worse fitness value with a certain probability. Therefore, the probability is calculated on the difference in fitness values between the new and current solution as defined by equation (9).

$$P(t) = e^{-\frac{\Delta f_t}{T}} \quad (9)$$

where T is the current temperature (scaling parameter),Δf_t is the difference in the values of the result between the current and the candidate solutions at step The temperature has an initial value T_o and it is reduced progressively according to a predefined cooling schedule α (see in Fig. 6); α is number between 0 and 1. As the temperature at t+1 iteration is calculated by equation (10).

$$T(t+1) = \alpha \cdot T(t) \quad (10)$$

The pseudo code of SA is shown as below.

Algorithm 3. Simulated Annealing

Initialization $\alpha, T, \varepsilon, K$
Randomly generate an initial state as the current solution C
while $T > Fuzzen$ **do** $k \leftarrow 0$
  while $k < K$ do **Generate** the candidate solutions N
    if $\Delta f_t > 0$ **then** $C \leftarrow N$
    else if $Random[0,1] > e^{-\frac{\Delta f_t}{T}}$ **then** $C \leftarrow N$
    $k \leftarrow k + 1$
  end while
  $T(t+1) = \alpha \cdot T(t)$
end while

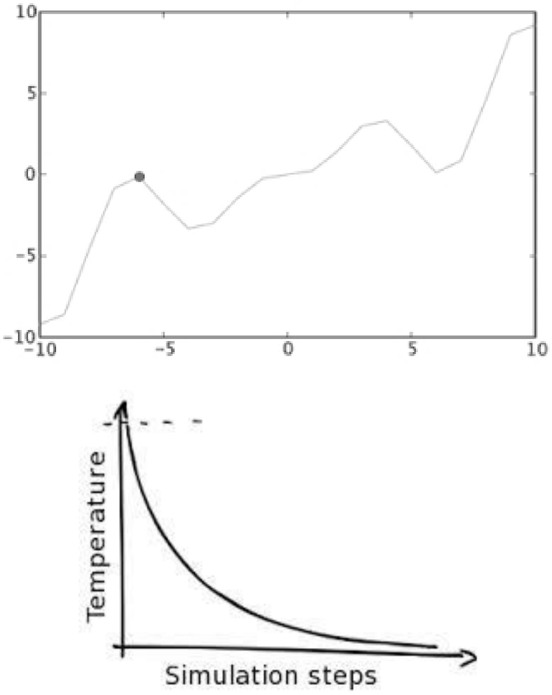

Fig. 6. Convergence graph of SA.

D. Framework of Discrete Radius Particle Swarm Optimization with Simulated Annealing for the MKPs

We deal a novel approach of Hybrid discrete radius particle swarm optimization and simulated annealing (H-DRPSOSA) to solve MKPs. The features of DRPSO and SA are fused to create an innovative approach, which can generate high-quality solutions in shorter calculation times and with more stable convergence characteristics. In the H-DRPSOSA, the global best position is selected from the agent particles of DRPSO in the search space. There are two ways of accepting a candidate solution. The H-DRPSOSA allows the fitness of some particles may be accepting a candidate solution with a worse fitness within a certain probability by the metropolis process of SA.

That note, the temperature is controlled the convergence graph of SA according to a predefined cooling schedule(see in Fig. 6). Addition, we propose the adaptive temperature is obtained by dividing the difference between the maximum and minimum fitness of the DRPSO. Thus, the initial temperature is changed in each MKPs dataset.The main point, the H-DRPSOSA finds the best solution and skips local optima by allowing the exploration of the problem space in the direction that leads to a local increase in the next solution. In the final state, after the system is cool, each particle in the swarm updates position using the best position from the hybrid state and then opens up to find the global optimum.

The H-DRPSOSA algorithm is elaborated below:

Algorithm 4. Framework of H-DRPSOSA

$T \leftarrow max\ fitness(x_{i,j}(0) - min\ fitness\ x_{i,j}(0)/ln06$
for each time step t to t_{max} **do**
  for each particle i in the swarm **do**
    if $rand() \geq S(v_{i,j}(t+1))$ **then** $x_{i,j}(t+1) \leftarrow$
  else $x_{i,j}(t+1) \leftarrow 1$
  end if
    calculate $f\left(x_{i,j}(t+1)\right)$
    if $f\left(x_{i,j}(t+1)\right) > pbest_i$ **then update** $pbest_i$
    find $abest_i$
    if $abest_i > gbest$ **then update** $gbest$
    else if $(T(t+1) > K)$ **then**
      current solution $C \leftarrow gbest$
      Generate the candidate solution N
      $\Delta f = f(C) - f(N)$
  end if
    if $\Delta f > 0$ **then** $gbest \leftarrow N$
    else if $Random[0,1] > e^{-\frac{\Delta f}{T}}$ **then** $gbest \leftarrow N$
  end if
  end for
end for

Step 1. Let the iteration t_{max}, weight w, the cooling rate α and system equilibrium K.

Step 2. Initialize position of the particles with randomly.

Step 3. Evaluate the fitness value of all particles which determine abest by the each particle.

Step 4. Determine the initial temperature T_0 with the difference between the maximum and minimum fitness of the initial particle in the swarm by the acceptance probability.

Step 5. Calculate the new position for each particle.

Step 6. If abest is better than gbest then update gbest and return to Step 5.

Step 7. If abest is less than gbest and T_0 more than frozen value ε, set the current solution C is $gbest$.

Step 8. Calculate the delta fitness Δf_t the candidate solution N is better than current solution C, update $gbest$ with the candidate solution N and return to Step 5.

Step 9. If the candidate solution N is not better than current solution C and the random number more than certain probability update $gbest$ with the candidate solution N and return to Step 5.

Step 10. If the evolution process not met the end criterion return to step 5.

Step 11. Output the best solution $gbest$ and its fitness value.

E. Solution Representation

To solve the MKPs, a candidate solution represent as the dimension is the number of items n, as shown in Fig. 7. For the example in Fig. 7, we have seven items and after the position of a particle is updated, the position representation is: 0.20, 0.90, 0.10, 0.70, 0.05, 0.65 and 0.80.

j_0	j_1	j_2	j_3	j_4	j_5	j_6
0.20	0.90	0.10	0.70	0.05	0.65	0.80

Fig. 7. Representation of the position in each dimension for MKPs.

F. Random Transfer Mechanism

The position update for the framework of H-DRPSOSA for the MKPs is taken from position update equation of DRPSO. If a random number value is more than the sigmoid value of the velocity then the position of particle takes the value 0. On the other hand, if a random number value is less than the sigmoid value of the velocity then the position of particle takes the value 1, as shown in Fig. 8.

$v_{i,j}(t+1)$	0.20	0.90	0.10	0.70	0.05	0.65
$S\left(v_{i,j}(t+1)\right)$	0.55	0.71	0.52	0.67	0.51	0.66
Random number	0.63	0.55	0.97	0.09	0.47	0.26
$x_{i,j}(t+1)$	0	1	0	1	1	1

Fig. 8. Representation of random transfer mechanism of positions for MKPs.

IV. Experiments and Results

The MKPs are taken from [23] (See in Table I). The benchmarks are selected from MP-Test-data SAC-94 suite.Therefore, the proposed method is tested using four different numbers of items (m x n). For RPSO, the size of swarm is 60, iteration number is 4000 or 240000 function evaluations, x_{max} and v_{max} are set to equal and within the range of [-4, 4], w is 0.98, μ is 0.4 (Scale factor of RPSO). For the SA, the cooling rate α is 0.95 and frozen ε is 0.001 the system equilibrium K is 1. Therefore, the comparison is made on the basic of optimum rate, best fitness, and average fitness.

TABLE I: The MKPs Dataset

Id	Dataset	m	n	Optimum (Opt)
1	Weish20	3	70	9450
2	Pb6	30	20	776
3	Sent1	30	40	7772
4	Sent2	30	60	8722

TABLE II: Comparison Optimum Rate of H-DRPSOSA, BPSO and SA Algorithm for MKPs Dataset

Dataset	Algorithm	Opt. (%)	Best	Average
Weish20	H-DRPSOSA	95	9450	9356
	BPSO	60	9450	9014
	SA	0	7787	7078
Pb6	H-DRPSOSA	90	776	760
	BPSO	15	776	652
	SA	5	776	642
Sent1	H-DRPSOSA	87	7772	7750
	BPSO	34	7772	7758
	SA	0	6939	6272
Sent2	H-DRPSOSA	86	8722	8610
	BPSO	5	8722	8112
	SA	0	8311	7995

TABLE III: Comparison Optimum Rate of H-DRPSOSA and Another Algorithm for MKPs

Dataset	Algorithm	Reference	Opt. (%)
Weish20	H-DRPSOSA	This study	95
	CBPSOTVA	Chih *et al*. [24]	78
	BPSOTVAC	Chih *et al*. [24]	69
	BPSO	Cho et at. [25]	53
Pb6	H-DRPSOSA	This study	90
	CBPSOTVA	Chih *et al*. [24]	50
	BPSOTVAC	Chih *et al*. [24]	54
	BPSO	Cho *et at*. [25]	28
Sent1	H-DRPSOSA	This study	87
	CBPSOTVA	Chih *et al*. [24]	57
	BPSOTVAC	Chih *et al*. [24]	39
	BPSO	Cho et at. [25]	16
Sent2	H-DRPSOSA	This study	86
	CBPSOTVA	Chih *et al*. [24]	27
	BPSOTVAC	Chih *et al*. [24]	2
	BPSO	Cho *et al*. [25]	3

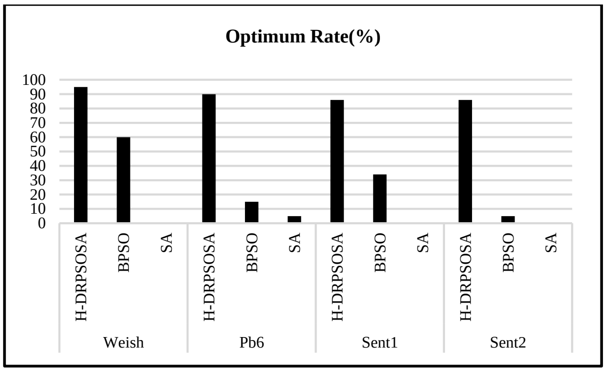

Fig. 9. Comparison optimum rate of H-DRPSOSA, BPSO and SA algorithm for MKPs dataset.

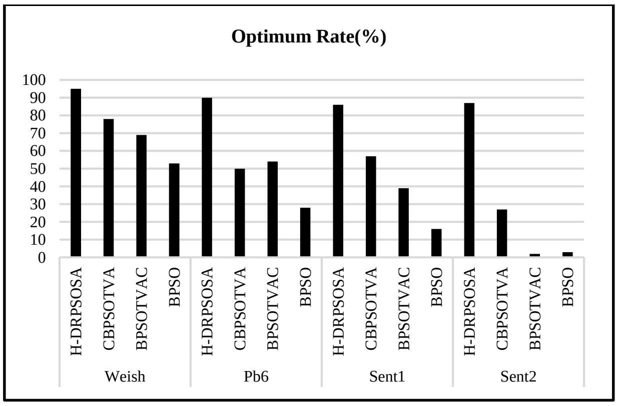

Fig. 10. Comparison optimum rate of H-DRPSOSA and another algorithm for MKPs.

Concerning the fitness function, it should note that in the MKPs, the fitness of each particle is related to the expected profit value of the selected items. Since the problem is to find the maximum profit. When the global optimum is achieved or the procedure reaches the maximum number of iterations, the selected items with maximum profit obtained by the H-DRPSOSA is returned as the result.

The experimental result of the MKPs datasets (in Table I) are illustrated in Table II-Table III. In Table II and Fig. 9 show the result of MKPs, it is clear that H-DRPSOSA is more reliable than BPSO and SA in term of optimum rate, best and average fitness. The result of MKPs, it is obvious that H-DRPSOSA performs better than BPSO and SA algorithm in all aspects (optimum rate and average fitness).

Regarding to optimum rate, the H-DRPSOSA outperforms the heuristic methods, which previously published in the literature under the same dataset. Therefore, it seem to be that Table III and Fig. 10 show the optimum rate increases as the all the test case. Moreover, all test case show that the H-DRPSOSA is very robust.

V. Conclusion

In this paper, the discrete radius particle swarm optimization (DRPSO) algorithm is modified with adaptive mutation for discrete domain, in which particles are regrouped within a given radius and the agent particle is determined, which is the best particle of the group for each local optimum. The DRPSO can maintain appropriate swarm diversity and jump out the local optimum using the agent particle to achieve the global optimum. Subsequently, the DRPSO can be combined with adaptive mutation to solve the impact on the maximum or minimum velocity in searching for the discrete regional solution, such that the diversity among particles moderately decreases. Also this paper provided a further test on the DRPSO algorithm using difficult well-known problems (Knapsack Problems) in order to verify its effectiveness and efficiency. By hybrid of the DRPSO and SA algorithm (H-DRPSOSA), the structure of our method employs the advantages of the DRPSO algorithm, including strong global search ability with those of the SA algorithm, which has a strong local search ability to obtain a good solution rapidly and accurately.Addition, we present the adaptive temperature of SA with the particle of DRPSO for solve each MKPs dataset. In conclusion, the performance of the H-DRPSOSA algorithm is proven using well-known large scale complex problems, the computational results of the proposed algorithm were compared with state-of-the-art heuristics. The result demonstrates that the proposed algorithm outperforms existing procedures in the literature in terms of optimum rate on the standard instance sets.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

A, B conducted the research; B analyzed the data; A, B wrote the paper; all authors had approved the final version.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful comments and suggestions. Research of the first author was supported by Rajamangala University of Technology Suvarnabhumi, Thailand.

References

[1] I. Boussald, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Information Sciences*, vol. 237, pp. 82-117, Jul. 2013.

[2] F. Maffioli, "Randomized algorithms in combinatorial optimization: A Survey," *Discrete Applied Mathematics*, vol. 14, no. 2, pp. 157-170, Jun. 1986.

[3] J. A. Parekp, A. Ruiz-Cortes, S. Lozano, and P. Fernandez, "Metaheuristic optimization frameworks: A survey and benchmarking," *Soft Computing*, vol. 16, pp. 527-561, Mar. 2012.

[4] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli, "Hybrid metaheuristics in combinatorial optimization: A survey," *Applied Soft Computing*, vol. 11, no. 6, pp. 4135–4151, Sep. 2011.

[5] A. Zhou, B. Qu, H. Li, S. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32-49, Mar. 2011.

[6] F. Gurski, C. Rehs, and J. Rethmann, "Knapsack problems: A parameterized point of view," *Theoretical Computer Science*, vol. 775, no. 5, pp. 93–108, Jul. 2019.

[7] K. Iwama and G. Zhang, "Online knapsack with resource augmentation," *Information Processing Letters*, vol. 110, no. 22, pp. 1016-1020, Oct. 2010.

[8] C. V. Henkel, T. Back, J. N. Kok, G. Rozenberg, and H. P. Spaink, "DNA computing of solutions to knapsack problems," *BioSystem*, vol. 88, no. 1-2, pp. 156-162, Mar. 2007.

[9] O. Granno and B. J. Oommen, "Learning automata-based solutions to the optimal web polling problem modelled as a nonlinear fractional knapsack problem," *Engineering Application of Artificial Intelligence*, vol. 24, no. 7, pp. 1238-1251, Oct. 2011.

[10] E. Bas, "Surrogate relaxation of a fuzzy multidimensional 0–1 knapsack model by surrogate constraint normalization rules and a methodology for multi-attribute project portfolio selection," *Engineering Application of Artificial Intelligence*, vol. 25, no. 5, pp. 958-970, Aug. 2012.

[11] J. Egeblad and D. Pisinger, "Heuristic approaches for the two-and three-dimension knapsack packing problem," *Computer & Operation Research*, vol. 36, no. 4, pp. 1026-1049, Apr. 2009.

[12] A. Pham, "The improvent knapsack chipper," *Computer Comunications*, vol. 34, no. 3, pp. 342-343, Mar. 2011.

[13] M. Fampa, D. Lubke, F. Wang, and H. Wolkowicz. (Aug. 2019). Parametric convex quadratic relaxation of the quadratic knapsack problem. *European Journal of Operational Research*. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0377221719306976

[14] F. Furini, M. Monaci, and E. Traversi, "Exact approaches for the knapsack problem with setups," *Computers & Operations Research*, vol. 99, pp. 208-220, Feb. 2018.

[15] F. Escobar, A. Kolar, N. Harb, F. V. D. Santos, and C. Valderrama, "Scalable shared-memory architecture to solve the Knapsack 0/1 problem," *Microprocessors and Microsystems*, vol. 50, pp. 189-201, May 2017.

[16] B. Thiongane, A. Nagih, and G. Plateau, "Lagrangean heuristics combined with reoptimization for the 0–1 bidimensional knapsack problem," *Discrete Applied Mathematics*, vol. 154, no. 5, pp. 2200–2211, Oct. 2006.

[17] J. C. Bansal and K. Deep, "A modified binary particle swarm optimization for knapsack problems," *Applied Mathematics and Computation*, vol. 218, no. 22, pp. 11042–11061, Jul. 2012.

[18] X. Zhang, S. Huang, Y. Hu, Y. Zhang, S. Mahadevan, and Y. Deng, "Solving 0-1 knapsack problems based on amoeboid organism algorithm," *Applied Mathematics and Computation*, vol. 219, no. 19, pp. 9959–9970, Jul. 2013.

[19] L. Wang, S. Wang, and Y. Xu, "An effective hybrid EDA-based algorithm for solving multidimensional knapsack problem," *Expert Systems with Applications*, vol. 39, no. 5, pp. 5593–5599, Apr. 2012.

[20] J. Kenney and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Conference on Neural Networks*, The Perth, The Australia, 1995, pp. 1942-1948.

[21] M. Anantathanvit and M. Munlin, "Radius particle swarm optimization," in *Proc. the Computer Science and Engineering Conference (ICSEC)*, Nakorn Pathom, Thailand, IEEE, 2013, pp. 121-130.

[22] S. Kirkpartrick, Jr. C. D. Gelatt, and M. P. Vecchi, "Optimzation by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671-680, May 1983.

[23] The Zuse Institute Berlin (ZIB). [Online] Avaliable: www.zib.de/index.php?id=921&no_cache=1&L=0.&cHash=fbd4ff9555f8714ac6238261e3963432&type=9*h*

[24] M. Chih, C. C.-J. Lin, M.-S. Chern, and T.-Y. Ou, "Particle swarm optimization with time-varying acceleration coefficients for the multidimensional knapsack problem," *Applied Mathematical Modelling*, vol. 38, no. 4, pp. 1338–1350, Feb. 2014.
[25] H. Cho, D. Kim, F. Olvera, and S. D. Y. Guilerma, "Enhanced speciation in particle swarm optimization for multi-modal problems," *European Journal of Operational Research*, vol. 213, no. 1, pp. 15–23, Aug. 2011.

E. Songkroh is a lecturer at the Faculty of Business and Information Technology, Rajamangala University of Technology Suvarnabhumi, Thailand. He got the bachelor's degree from the Faculty of Engineering, Industrial Engineering, master of science program in information technology, Rangsit University, Thailand. Currently, he teaches the subject relate to computer science. His research interests are data mining and computer vision.

M. Anantathanavit received Bachelor degree in IE Tech. from Faculty of Industrial Education and Technology, King Mongkut's Institute of Technology Ladkrabang (KMITL), master of business administration (M.B.A) and master of science program (M.S.) in information technology from Ramkhamhaeng University (RU), and Ph.D. in information technology from information science and technology, Mahanakorn University of Technology (MUT), Thailand. His research interests are metaheuristic, swarm intelligence, artificial intelligence, and optimization algorithm.